Langages & systèmes

LE MAGAZINE DE LA MICRO-INFORMATIQUE PROFESSIONNELLE

MAI 1992 - N° 7
SUPPLÉMENT
DÉVELOPPEURS
DU N° 81
D'INFOPC
ISSN 0951- 6402

APPRENDRE WINDOWS

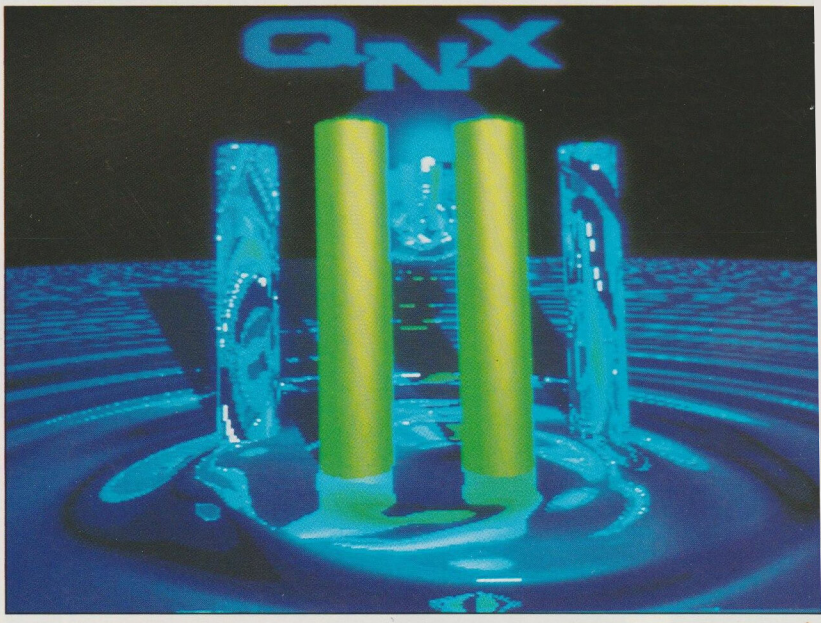
Présentation des polices bitmap et des notions d'impression

LES TEMPLATES DE BORLAND C++

Concevoir facilement des bibliothèques de classes

FICHIERS BMP

Analyse des caractéristiques des fichiers bitmap de Windows 3.0



LES OS TEMPS RÉEL

Etude des systèmes d'exploitation temps réel compatibles MS-DOS

LE FORMAT TGA

Présentation de la structure d'un fichier Targa

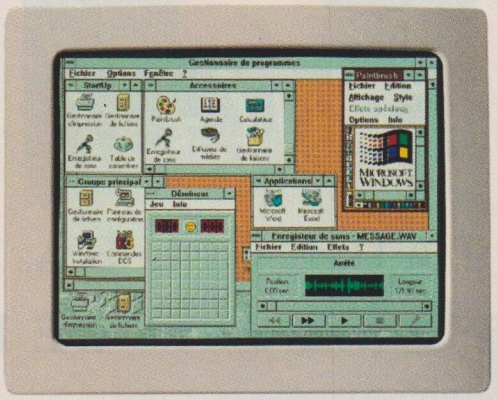
ET AUSSI...

- LES ALGORITHMES DE TRI
- ZORTECH C++
- LES APPORTS DE dBASE IV 1.5
- PÉRISCOPE

NE PEUT ÊTRE VENDU SÉPARÉMENT

Nouveau Microsoft Windows 3.1 Là, ça va très loin.

Attachez vos ceintures... Votre PC s'apprête à décoller vers de nouvelles cimes. Là où la convivialité règne en maître, où la puissance des applications s'exprime sans réserve.
Avec Microsoft Windows 3.1, découvrez de nouveaux espaces de liberté, dans un environnement où tout est plus simple et rapide. Exploitez plusieurs applications simultanément. Réalisez très facilement des documents d'une sophistication élevée grâce à la fonction OLE de Windows 3.1. Agrémentez-les des polices de caractères TrueType qui amènent une véritable similitude entre l'écran et le papier et imprimez plus rapidement. Réorganisez plus aisément vos informations grâce au "glisser-déplacer" à la souris. Découvrez les vertus du multi-média en accueillant le son et l'animation dans vos applications d'aujourd'hui et celles de demain. Avec les extensions Windows pour Pen Computing*, profitez des dernières technologies. Bienvenue à bord du vaisseau Windows 3.1. Il va vous faciliter la vie de mille manières.



Windows 3.1, tonifiez vos logiciels. Tout est plus rapide sous Windows 3.1. Les logiciels disponibles - plus de 5000 - semblent avoir pris des vitamines. La vitesse d'accès disque est améliorée de 50%, l'impression de 25%, l'affichage paraît instantané. Les polices de caractères TrueType apportent une véritable similitude entre l'écran et le papier.
La nouvelle gestion mémoire permet d'exécuter deux fois plus d'applications que par le passé. Windows 3.1 améliore même les applications MS-DOS: elles apparaissent dans des fenêtres et sont sensibles à la souris.

Suivez le guide! L'environnement graphique des années 90 est accessible à tous. L'installation s'effectue automatiquement: Windows 3.1 analyse votre PC et se configure en conséquence. Les débutants se familiariseront avec le système en lançant le didacticiel. A tout moment, chacun peut faire appel à une aide adaptée à l'opération en cours.

Performant et simple. Windows 3.1 est extrêmement fiable: plus de 15 000 sites l'ont testé avant même sa disponibilité. Des années de recherche auprès de nos 10 millions d'utilisateurs ont rendu son utilisation plus simple (plus de 1000 améliorations). Le "glisser-déplacer" à la souris effectue en une manipulation ce qui en demandait trois auparavant.
La fonction OLE permet de travailler sur des documents composés de données provenant de plusieurs applications.

L'avance technologique de Microsoft. Windows 3.1 intègre les technologies du futur. Les fonctions multi-média de Windows 3.1 enrichissent en son et en image vos documents de tous les jours. L'extension Windows pour Pen Computing*, adaptée aux ordinateurs sans clavier ouvre la bureautique à de nouveaux utilisateurs.

Pour toute information complémentaire:

3616 Microsoft
pour consulter la liste des revendeurs agréés.

Le Service Clients: 16 (1) 69 29 11 11

Microsoft: Infomart CNIT - La Défense

MICROSOFT - 91957 LES ULIS CEDEX

Service lecteurs n° 3 (dernière page)

* disponible séparément.

ROUX SEGUELA CAYZAC & GOUDARD

Microsoft®

LES LOGICIELS QUI DONNENT DES AILES.

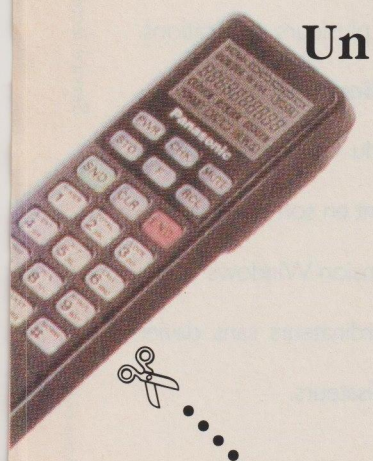
Service lecteurs n° 4 (dernière page)

TAU CETI S.A.

LES TECHNOLOGUES DE L'OBJET

SMALLTALK/V®

Un développement de 5 à 10 fois plus rapide sous OS/2 PM et Windows



SMALLTALK/V PM
SMALLTALK/V WINDOWS

PRIX UNIQUE
2 670 FRS T.T.C.
TVA 18,6 % - Franco de Port
Pour toute commande supérieure à 10 nous consulter

SMALLTALK/V est une marque déposée de Digitalk Inc, 8éme membre d'IBM International Alliance pour AD/Cycle.

OFFRE PRIVILEGE

A retourner à TAU CETI 13, rue de Lens 92000 NANTERRE

Mr Mme Melle _____ Fonction _____
Société _____ Ville _____
Adresse _____ Fax : _____
C.P. _____
Tél. : _____
OS/2PM Nbre : _____
Windows Nbre : _____
Réglement par chèque à joindre à la commande

TAU CETI S.A. - 13, rue de Lens 92000 Nanterre
Tél. : 47 80 67 61 - Fax : 47 80 29 21
S.A. au capital de 5 000 000 Frs

Toutes les marques citées sont la propriété de leurs dépositaires respectifs

Communik

Sommaire

- 9 -

Re-sources

Une sélection des nouveaux outils de développement.

- 16 -

Les templates de Borland C++ 3.0

Les templates facilitent la conception de bibliothèques de classes.

- 22 -

Apprendre Windows

Suite et fin de l'étude de l'interface GDI : présentation des polices bitmap et des notions d'impression.

- 44 -

Structure des fichiers BMP

Analyse des caractéristiques des fichiers bitmap de Windows.

- 48 -

Les OS temps réel

Peut-on à la fois faire du temps réel et être compatible MS-DOS? Le point sur six systèmes d'exploitation.

- 70 -

Les apports de dBase IV 1.5

Borland a revu et corrigé dBase IV. Ce dernier est désormais plus fiable et s'exécute plus vite.

- 76 -

Les algorithmes de tri

Avantages et inconvénients des différents algorithmes de tri.

- 88 -

Periscope

La carte Periscope permet de réaliser des débogages très poussés, au-delà des solutions logicielles classiques.

- 96 -

Le format TGA

Présentation de la structure d'un fichier Targa.

- 102 -

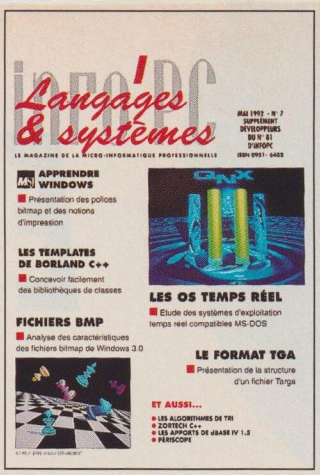
Zortech C++

Face aux compilateurs C++ de Microsoft et Borland, celui de Symantec, le Zortech C++, n'est pas dénué d'atouts.

- 108 -

Concours GOLD

Vous avez conçu une application pour un usage professionnel? Faites-la découvrir au jury du Grand Open Logiciel et Développement organisé par Langages & Systèmes.



La révolution... pour après-demain

Dans un article de septembre 1991 de la revue Scientific American, Mark Weiser, directeur du laboratoire des ordinateurs de Xerox au PARC (Palo Alto Research Center), affirme que l'ordinateur ne pourra entrer dans les domaines qu'il n'a fait qu'effleurer que lorsqu'il se sera fondu dans l'environnement. Il explique en détail les solutions qu'il a développées pour obtenir cette «fusion». Elles sont toutes basées sur un ordinateur «qui se cacherait derrière son écran» pour devenir une «feuille de papier intelligente». Tous les éléments nécessaires à la production de cette machine existent. Reste à les réunir.

Grid et NCR commercialisent depuis quelque temps des tablettes graphiques qui sont les «ancêtres» de cette feuille intelligente. Mais ces sociétés ont péché par leurs choix techniques : l'écran de ces machines demeure illisible dans la plupart des situations en raison de la technologie employée (écran LCD à matrice passive), les microprocesseurs retenus sont lents pour un traitement en environnement graphique, et l'équipement en mémoire vive et en mémoire de masse ne donne pas satisfaction. Cependant, le fait de changer une technologie d'écran par une autre et d'augmenter la mémoire ne suffit pas : sans applications, la feuille intelligente sera une invention inutile. Or le développement des dites applications connaît des difficultés importantes, tant techniques que «philosophiques».

Les problèmes techniques sont bien connus. En revanche, la conception d'un logiciel pour une telle plate-forme soulève des interrogations : un simple «portage» suffit-il, au risque de sous-exploiter les caractéristiques de la machine, ou faut-il définir une nouvelle interface avec sa propre SAA, impliquant un consensus de la part des principaux éditeurs? Cette unanimité sera certainement aussi difficile à obtenir que la résolution des problèmes techniques. Autrement dit, les ordinateurs de bureau et les portables ont encore de beaux jours devant eux...

Yves Guittard

DEVELOP

Version 100% linkable!

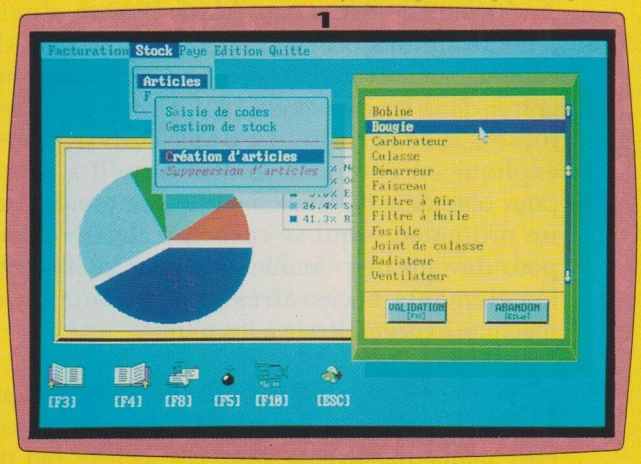
HIGH SCREEN 5.5

Générateur d'écrans
- DOS, WINDOWS, OS/2, UNIX.
- Tous langages (oui, tous!)

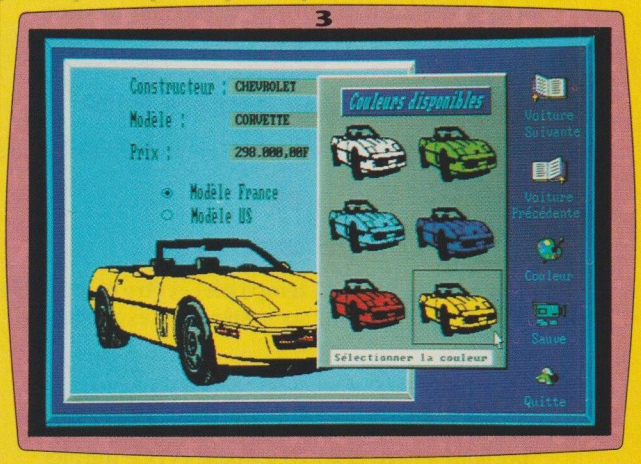
Nouvelles versions:
- **UNIX**
- **Windows**

1- Menus déroulants, fenêtres, saisies, icônes, graphiques,... : tout High Screen 5 en un écran !
2- High Screen fonctionne sur tous les écrans...
3- Une application graphique sous DOS...
4-... la même sous Windows (compatibilité !)

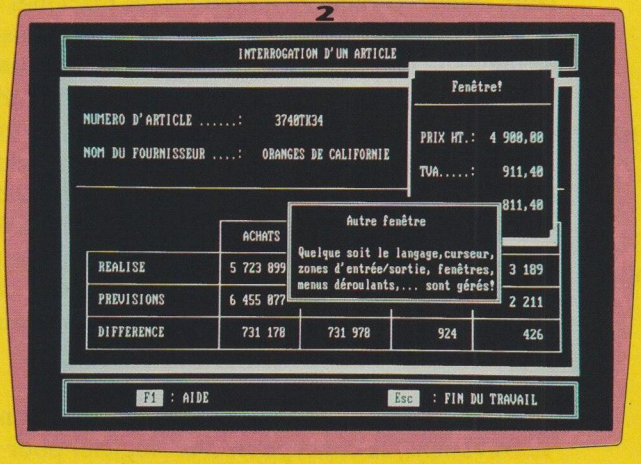
1



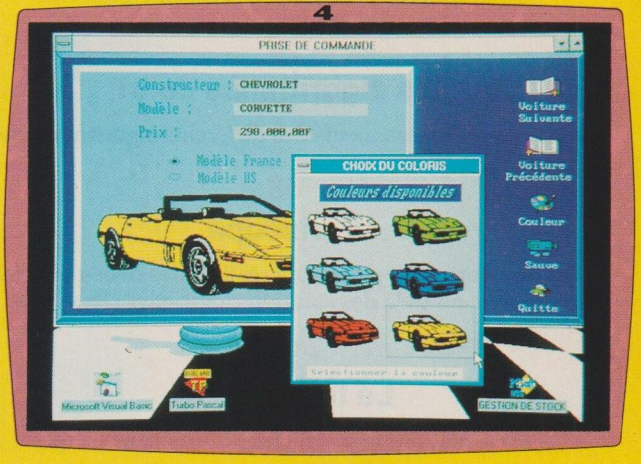
3



2



4



Que vos applications soient en mode texte ou graphique, au clavier ou à la souris, avec menus déroulants, avec des icones, avec ou sans fenêtres,
High Screen 5 vous permettra de développer 3 à 10 fois plus vite.

La programmation est très simple, avec des ordres en français.
De nombreux utilitaires sont livrés : éditeur d'icones, de dossiers, maquettage,... Tout devient possible, dans tous les langages!
(C, Basic, Pascal, dBase, Nantucket, Cobol, Fortran,...)

High Screen 5 a déjà changé la vie de milliers de développeurs : essayez-le !

A partir de
4900 F HT
5 811,40 F TTC

Disquette de démonstration disponible : appelez !

Nos produits ont reçu de nombreuses récompenses :

HYPER PACK — OUTILS DE DEVELOPPEMENT — PRIX D'EXCELLENCE — micro systèmes

Meilleur outil de développement — SOFT & MICRO

LABO QUALIFIE — INFO PC

LE SUPPORT TECHNIQUE
Un des points forts de PC SOFT est le support technique, efficace, disponible et... sympathique.
Notre équipe de techniciens chevronnés est à votre disposition. Gratuitement.

GARANTIE "ESSAI SANS RISQUE"
Si le produit ne vous convenait pas pour une raison quelconque, vous pourriez nous le retourner (en état d'origine bien sûr) dans les 10 jours suivant sa réception pour un remboursement intégral. Conditions au dos du tarif.

Malgré le soin apporté à la rédaction de ce document, les informations ne sont pas contractuelles. Toutes les marques sont déposées par leur propriétaire respectif.

PEZ VITE

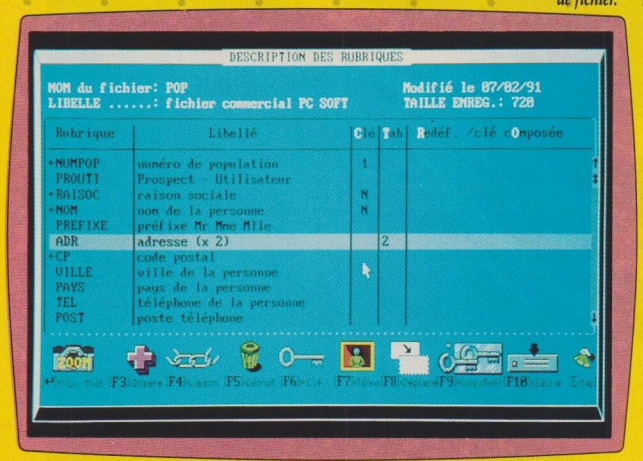
Version 100% linkable!

HYPER FILE 3

SGBD Gestionnaire de fichiers

- DOS - WINDOWS - OS/2
- C - Basic - Pascal

Création de structure de fichier.



Hyper File 3 est le plus rapide des SGBD. Après avoir défini facilement la structure des fichiers, leur utilisation dans les programmes est très simple.
La programmation (en C, Basic, Pascal) s'effectue en français.
De plus Hyper File 3 édite automatiquement le dossier d'analyse.
De nombreux utilitaires sont livrés.

Si vous manipulez des fichiers, testez Hyper File !

A partir de
4900 F HT
5 811,40 F TTC

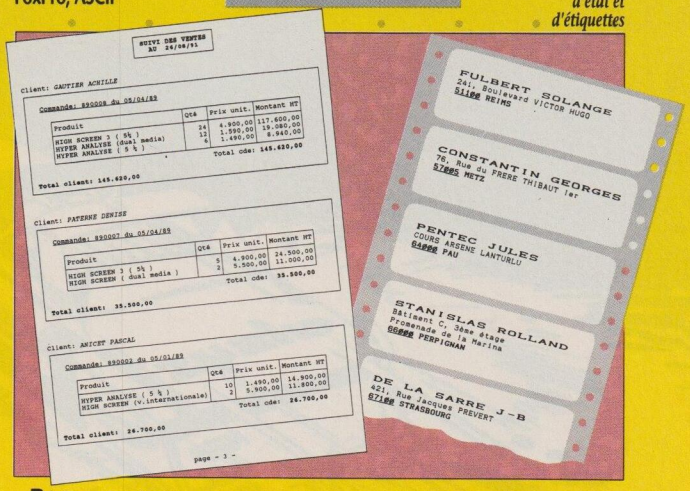
Disquette de démonstration disponible : appelez !

HYPER PRINT 3

Générateur d'états et d'interrogations

- C, Basic, Pascal, dBase, Nantucket, FoxPro, ASCII

Exemple d'état et d'étiquettes



Réaliser des états imprimés de gestion en quelques minutes seulement, c'est possible avec Hyper Print 3.
Hyper Print 3 rapproche jusqu'à 9 fichiers par état et sait calculer.
Le résultat s'édite sur imprimante (matricielle ou laser) ou à l'écran, ou même dans une feuille de tableur.

Un produit à évaluer d'urgence !

A partir de
4900 F HT
5 811,40 F TTC

Disquette de démonstration disponible : appelez !

HYPER PACK DEVELOPPEUR

Atelier de Génie Logiciel

C - Basic - Pascal

Ce "pack" regroupe High Screen, Hyper File et Hyper Print qui utilisent le même dictionnaire de données.
Le "PACK" permet de développer de superbes applications rapides, très très vite !

A partir de
9900 F HT
11741,40 F TTC

PC SOFT est Fournisseur Officiel de la Préparation Olympique

PCSOFT

LIVRAISON RAPIDE
En France métropolitaine, livraison en 24 heures par transporteur.

DISQUETTE DE DEMONSTRATION DISPONIBLE

DOCUMENTATIONS CLAIRES EN FRANCAIS

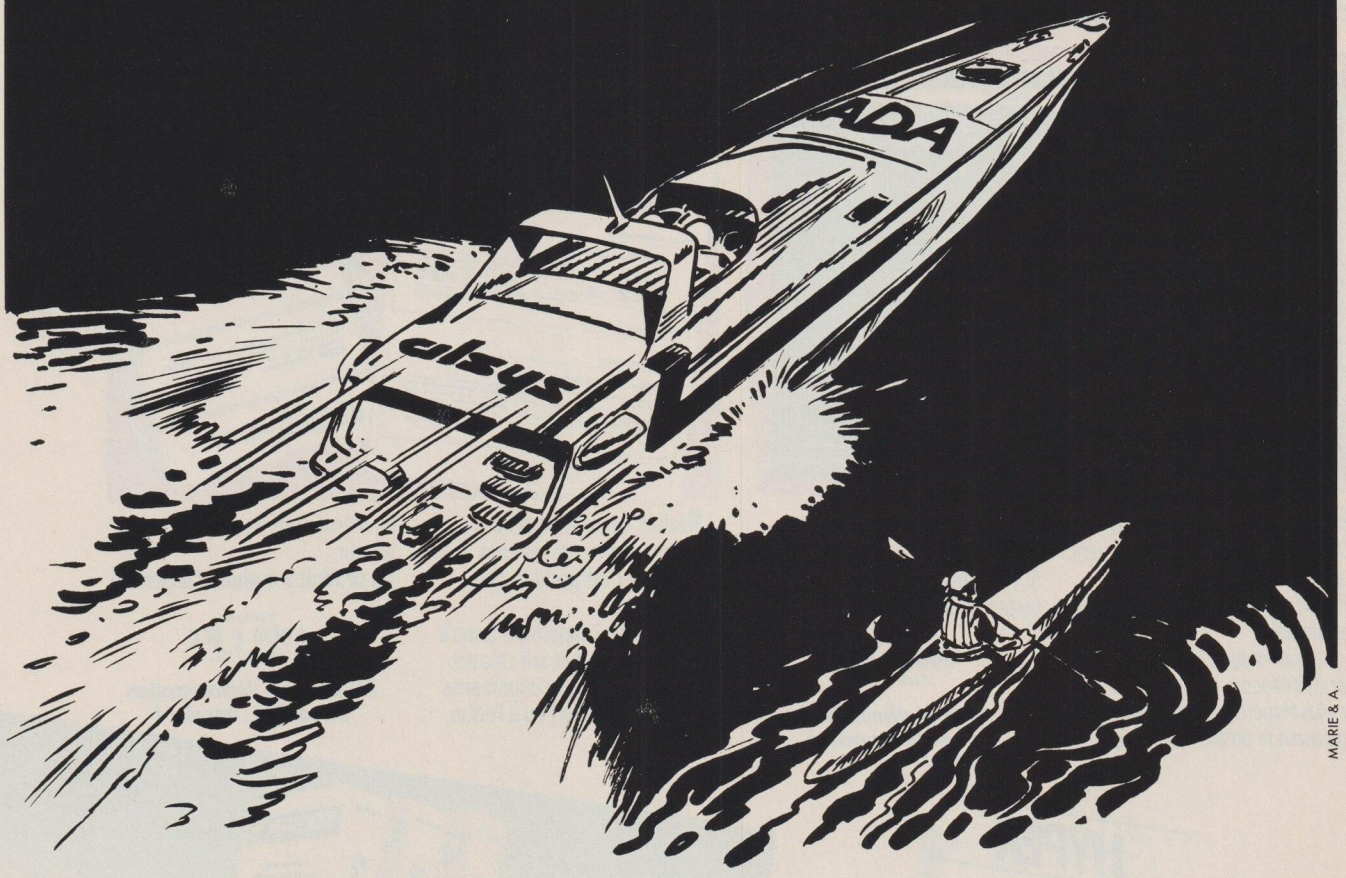
PAS DE REDEVANCES

SIEGE MONTPELLIER
Tél. (16) 67.032.032
216, rue des Escarceliers, BP 3019
34034 Montpellier Cedex 01
Fax: (16) 67.03.07.87
Support Technique: (16) 67.03.17.17

PARIS
Tél. (1) 48.01.48.88
34, Boulevard Haussmann
75009 Paris
Télex: 290.266 F (MBI)
Minitel: 3614 code PCSOFT

Service lecteurs n° 5 (dernière page)

Alsys Ada.
Le génie logiciel va faire des vagues.



MARIE & A.

Service lecteurs nº 6 (dernière page)

Espace, transports, robotique, contrôle industriel, forage, télécommunications, avionique, gestion... Dans tous les secteurs d'activité et tous les pays du monde, Alsys Ada est devenu la référence absolue des plus grandes entreprises.
Alsys, premier fournisseur mondial de produits Ada, vous offre le meilleur du génie logiciel, et le meilleur du temps réel. La puissance du langage Ada alliée à la richesse des environnements de développement Alsys est la clé d'un univers où la production industrielle de logiciels découvre une efficacité encore ignorée. Optimisation dans l'organisation de projet, capacité de développement distribué, facilité d'intégration, possibilité de réutilisation des composants, portabilité des programmes d'un environnement matériel à l'autre, efficacité du code généré, simplicité de maintenance, autant de qualités au service de la productivité.
Alsys Ada : une technologie alliant qualité et maturité qui a déjà fait ses preuves pour les projets les plus exigeants. Une technologie qu'Alsys met aujourd'hui à la portée de tous.

alsys
Groupe Thomson-CSF

29, avenue Lucien-René Duchesne 78170 La Celle Saint-Cloud
Tél. : (1) 39.18.12.44 - Fax : (1) 39.18.26.80

Re-sources

C++ 7.0

Microsoft se lance dans la programmation orientée objet. Avec le nouveau compilateur de Microsoft, la POO franchit une nouvelle étape.

En effet, le C++ 7.0 présente plusieurs innovations de taille concernant la définition des classes objets, la génération de code, et l'environnement de programmation. Tous ces éléments destinent ce produit au professionnels du développement sous Windows.

C++ 7.0

• Edité et distribué par Microsoft • Compatible ATT 2.1 et ANSI • Classes objet Dos et Windows • Editeur de ressources • Optimisation de la taille du code • Gestionnaire d'overlay • Source Profiler en standard • Nouvel version de l'environnement •

Dans le domaine de la portabiblité et de la compatibilité du langage, la version 7.0 est une fidèle implémentation du C++ ATT 2.1 et du C ANSI, à laquelle se joint de nouvelles techniques d'optimisation. Par exemple, le "Packcode", et la génération en ligne de fonctions, assurent le meilleur compromis entre la taille et la rapidité du code.

Outre les avantages de la POO, le C++ 7.0 propose une nouvelle génération de classes objet, encapsulant l'ensemble de l'interface de programmation de Windows, et cela avec la même puissance que le langage C.

L'utilisateur dispose d'un éditeur de ressources pour cet intégrateur graphique. Cela concerne les boîtes de dialogue, les icônes, les polices de caractères, et l'aide en ligne. Le compilateur propose un traitement des erreurs plus efficace, par la prise en compte des exceptions.

La création de modèles, "templates", contribue également à une meilleure finition. Le support du mode 32 bits conviendra aux applications les plus complexes. L'utilisateur trouvera trois outils dans le "Programmer's Workbench".

Le "Browser" permet l'examen interactif des relations entre les classes d'objet, les fonctions s'y rattachant et leurs données. Par exemple, les types imbriqués dans une classe sont locaux à cette classe.

De même, "Move", le nouveau gestionnaire d'overlay améliorera les performances des applications Dos, à travers leurs divers fichiers de recouvrement.

Cela comprend la gestion du "Swapping" sur disque, de la mémoire cache. Enfin, Microsoft a intégré en standard le "Source Profiler", outil offrant une analyse plus fine du code source.

Des extensions pour Visual Basic

Microsoft lance sur le marché le kit de développement complémentaire de Visual Basic qui apporte une vaste gamme d'outils destinés à la réalisation d'applications professionnelles.

Ce kit regroupe une vingtaine de contrôles personnalisés permettant de mieux intégrer les applications écrites en Visual Basic dans l'architecture Windows. Ces outils comportent un compilateur d'aide au format

Kit de développement pour Visual Basic

• Edité et distribué par Microsoft • Kit de développement complémentaire de Visual Basic • Compilateur d'aide - Création de procédures d'installation • Nombreux exemples • Bibliothèque d'images • Gestion du son et des images •

Interface NI-488.2

• Edité et distribué par Sacasa • Interfaçage logiciel entre Visual Basic et le NI-488.2 Windows • Programme de pilotage de carte IEEE-488.2 • Mesure physique et automatisation comme domaines d'applications •

Windows, le kit de développement et de débogage de Visual Basic, l'ensemble de création de procédures d'installation, des extraits de la base de connaissance OnLine, des exemples d'applications complètes développées avec Visual Basic (gestion des ports de communication, gestion d'un tableau de plus de 64 ko à l'aide d'une DDL, etc.). Il inclut de surcroît une bibliothèque complémentaire d'images, au format Windows Metafile (graphiques vectoriels).

Les contrôles personnalisés du kit de développement pour Visual Basic ont pour objectif de faciliter le développement d'applications pour Windows. Ils permettent d'intégrer des fonctions élaborées (tableaux, graphes, objets) et de gérer des sons ou des images.

La société Sacasa annonce de son côté la disponibilité de son interface NI-488.2. Cette interface assure la liaison entre un programme d'applications écrit sous Visual Basic et le logiciel de contrôle-commande NI-488.2 Windows de National Instruments. Ce dernier est fourni avec les cartes d'interface GPIB (IEEE 488.2) pour PC/XT/AT/EISA et IBM PS/2 que propose National Instruments. Il est donc possible de développer des applications GPIB sous Windows 3 en profitant des outils graphiques et du langage Visual Basic.

Quant aux domaines d'applications, ils s'étendent du test de production à l'automatisation, en passant par la recherche et le développement, partout où il est nécessaire d'utiliser des instruments programmables GPIB.

Une boîte à outils de développement photo sur CD

Kodak présente la boîte à outils de développement Photo CD. Cette bibliothèque de développement permet aux éditeurs de logiciels et aux intégrateurs de bénéficier de manière transparente dans leurs logiciels de la technologie de Photo CD Kodak, standard émergeant pour images numériques de haute qualité.

Cette boîte à outils est disponible pour Windows 3.0 et les environnements Mac. Le système Photo CD Kodak repose sur un disque compact ré-inscriptible qui enre-

Boîte à outils Photo

• Edité et distribué par Kodak • Boîte à outils de développement photo CD • Intégration d'image de qualité photo • Gestion de la décompression des fichiers images •

gistre sous un faible volume les images couleurs en format numérique, tout en conservant la résolution de la pellicule photographique.

La technologie du Photo CD aplanit les obstacles qui ont freiné l'utilisation massive d'images dans l'informatique de bureau : la qualité de l'image et le coût. En effet, la boîte à outils Photo CD, permet aux développeurs d'intégrer sans effort les images Photo CD à leurs applications.

De plus, elle offre des fonctions pour construire une interface souple entre les applications et le système Photo CD, et gère automatiquement la décompression des fichiers images et leur conversion aux formats les plus répandus, les attributs de l'image tels que son orientation, sa définition, et la zone affichée.

Elle utilise le codage couleur photo YCC de Kodak qui convertit rapidement de données au format des moniteurs RVB et une donne accès à une utilisation performante des couleurs 24 bits.

ADA pour Station SPARC

Alsys met à disposition prochainement un environnement de développement croisé Ada pour les stations Sparc vers des machines équipées d'un coprocesseur de la gamme 680 x 0 ou CPU 32. Cet environnement de programmation est particulièrement destiné au développement des applications temps réel embarquées.

Le produit comprend un compilateur croisé avec plusieurs niveaux d'optimisation, un relieur et d'un environnement multibibliothèque. De plus, l'utilisateur dispose d'un outil de transfert, d'un module exécutif AdaExec, et du spport du noyau temps réel Adakernel. Pour ce qui est de l'interface utilisateur, le kit est livré avec AdaWorld commun à tous les produits Alsys. L'executif AdaExec et le noyau temps réel AdaKernel peuvent être mis en ROM indépendamment de l'application.

Les services d'AdaKernel qui réalisent les fonctionnalités CIFO sont accessibles depuis le programme Ada : sémaphores, identification des tâches, suspension ou activation des tâches, masquages ou démasquage des interruptions.

Le compilateur génère du code pour la gamme Motorola 680 x 0 ou pour la famille Motorola CPU 32 sous Ada Kernel (noyau temps réels propriétaire d'Alsys).

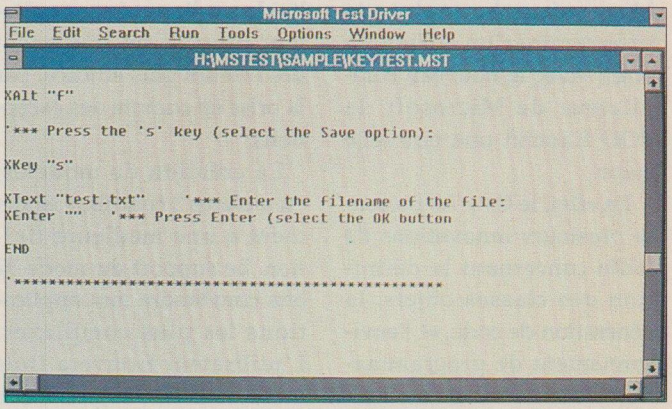
Bien adapté au langage Ada, il ordonnance les tâches sur la base de priorités avec préemption et un temps de réponse déterministe. Les programmes peuvent s'exécuter en mode utilisateur ou superviseur.

AdaKernel accepte des interruptions persistantes ou volatiles avec un temps de latence constant. Les entrées d'interruptions peuvent comporter des paramètres et être appelées depuis un programme.

ADA

• Edité et distribué par ALSYS • Environnement de développement ADA pour station Sparc • Génération de code 680 X 0 ou CPU 32 • Noyau temps réel • Gestion du multitâche •

Microsoft Test pour Windows



Des procédures de tests automatiques sous Windows.

Microsoft Test pour Windows automatise des tests d'applications s'exécutant sous Windows.

Cet outils manquait cruellement aux professionnels du dévelopement pour cet inétrageur graphique. Il permet de créer et d'exécuter des procédures de tests automatiques afin de contrôler le fonctionnement des applications Windows lors de la phase de mise au point.

Ce produit se destine aussi bien aux services informatiques des entreprises pour le test d'applications en interne qu'aux sociétés de service souhaitant accroître la fiabilité de leurs applications Windows avant leur commercialisation. Microsoft Test pour Windows contient des outils de tests prêts à l'emploi, modules exécutables qui vérifient les fonctionnalités logiques et graphiques des applications, et un ensemble de bibliothèques modulaires regroupant des fonctions de tests.

Il intègre également un langage interprété, proche de Visual Basic, pour la création de procédures de tests. Il est également possible d'utiliser les fonctions des bibliothèques DDL du Visual Basic, mais aussi des langages plus traditionnels tels le Microsoft C, le macro langage de Microsoft Excel, Borland Object Vision, Borland Turbo Pascal et C++ standard. Le développeur peut tester des évènement spécifiques de l'environnement Windows, comme l'envoi de messages entre applications, et détecter des évènement imprévus comme les erreurs non récupérables. Test pour Windows peut également être utilisé pour l'initialisation d'actions prédéfinies, comme le lancement d'un autre test et intègre un enregistreur qui génère du Basic à partir des actions claviers et souris. Ainsi l'historique des commandes sera toujours sauvegardé.

Microsoft Test

• Edité et distribué par Microsoft • Automatise les tests d'applications Windows et vérifie le fonctionnement logique et graphique des applications. • Peut être utilisé avec les langages de programmation sous Windows • Détecte des évènements imprévus comme les erreurs non récupérables •

Re-sources

Nouvelle version d'Omnis Mac et PC

La version 7 d'Omnis intègre un système de gestion de fichiers et un générateur d'application, et permet la définition de fichiers avec des structures relationnelles ou hiérarchiques dans un environnement entièrement graphique, en mode de développement et en mode utilisation. Le passage instantané entre ces deux modes accélère la mise au point des procédures.

Quant aux connexions avec SQL, Omnis 7 supporte de nombreuses bases de données, en plus de la possibilité pour le développeur d'incorporer des requêtes SQL correspondant au type de la base installée, l'interface SQL d'Omnis permet d'utiliser des mots-clé. Ceux ci servent à créer des tables SQL à partir de rubriques Omnis, à envoyer et à retrouver des informations sur le serveur et à accéder au dictionnaire de données SQL.

Les possibilités offertes par le DDE (Data Dynamic Exchange) facilitent l'intégration d'autres logiciels Windows. Omnis 7 peut aussi stocker des rubriques graphiques dans des bases SQL. L'interface Omnis à cinq fonctions principales : transmettre des ordres SQL situés au sein des programmes Omnis, communiquer avec la couche SQL pour laquelle elle a été créée, envoyer des requêtes SQL et en recevoir le résultat, placer les données résultantes dans des rubriques Omnis, convertir les données provenant de la base SQL en données Omnis et réciproquement, et enfin interpréter les mots-clé se trouvant dans les programmes Omnis de façon à générer des requêtes SQL compréhensibles par la base attaquée.

Omnis 7

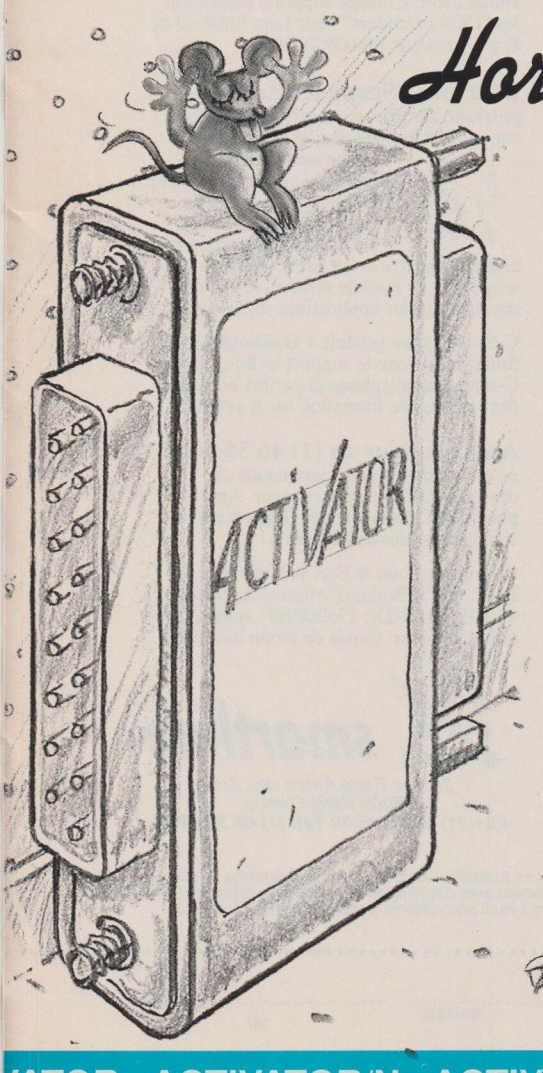
• Edité par Blyth Software • Distribué par Aware • Disponible sous Windows et Macintosh • Gestion de fichiers • Générateur d'application • Connexion SQL indépendamment de la base attaqué •

Une série de cartes graphiques sur-puissantes

Les cartes Vitec, conformes à la norme ANSI, sont de puissants calculateurs dédiés au traitement d'image pour les plates-formes UNIX. Ce type de matériel est destiné aux applications graphiques de haut niveau, telles que la simulation professionnelle, la télédétection, la cartographie, ou encore la recherche médicale. Les environnements C et X Windows, avec les systèmes hôtes SUN, HP, SGI, DEC et IBM sont supportés.

Ces produits se présentent sous la forme de cartes, totalement indépendantes de

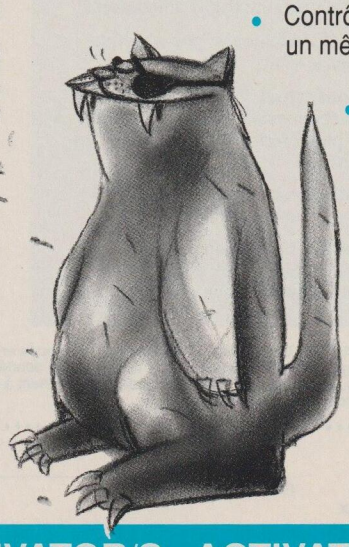
Hors de portée des convoitises !!!



SÉLECTIONNÉE PAR MICRO SIGMA LA GAMME ACTIVATOR

REPOND A VOS BESOINS SPECIFIQUES DE PROTECTION INFORMATIQUE

- Contrôle d'accès à des programmes et/ou modules par mots de passe
- Contrôle du nombre d'utilisateurs pouvant simultanément accéder à un même programme
- Contrôles programmables à partir d'informations stockées dans une mémoire non volatile
- Environnements : DOS, UNIX, MAC, SUN...
- A partir de 250 Frs. H.T.



Consultez-nous au 46 22 99 88 Fax : 47 63 84 42

MICRO SIGMA Outils de Développement
42, rue Legendre - 75017 PARIS

ACTIVATOR est une marque déposée de SOFWARE SECURITY - USA

Service lecteurs n° 7 (dernière page)

VATOR ACTIVATOR/N ACTIVATOR/S ACTIVATOR/M MACTIVATOR ACTIVATOR ACTI

C-scape : un véritable plaisir.

Gestionnaire d'interfaces écran graphique et texte

Le système de gestion d'écrans C-scape® est une bibliothèque de fonctions en langage C permettant la saisie et la validation de données à l'écran, la gestion de menus et fenêtres, l'édition de texte, et les systèmes d'aide en ligne. L'outil de conception d'écrans de C-Scape, Look & Feel™, vous permet de créer des écrans de façon interactive et de générer le code source correspondant.

C-Scape comprend des fonctions de haut niveau facilement modifiables, aussi bien que des primitives permettant de construire de nouvelles fonctions. Grâce à l'architecture orientée objet de C-Scape, vous développerez des applications plus fonctionnelles, plus flexibles, plus portables, et vous aurez plus de plaisir à le faire.

Un produit phare dans l'industrie du logiciel. Depuis 5 ans, plusieurs milliers de développeurs dans le monde entier se sont convertis à C-Scape, abandonnant leur bibliothèque "maison", ou des produits peu flexibles. La presse approuve: "C-Scape est de loin le meilleur. ...un véritable plaisir", ecrit IEEE Computer. Les plus grandes sociétés ont selectionné C-Scape comme standard de développement.

C-Scape peut être utilisé avec des SGBD, des bibliothèques graphiques, ou toute bibliothèque C ou C++. De plus, C-Scape peut combiner texte et graphique. Pour porter votre application de DOS ou OS/2 vers UNIX, VMS, AIX, recompilez simplement. C-Scape supporte également les "DOS extenders" Phar Lap, Rational et Ergo, ainsi que MetaWINDOW.

Graphique. Fonctionne en couleur, en mode texte ou graphique. Peut lire des images graphiques depuis des fichiers PCX.

Architecture orientée objet. Ajoutez vos propres fonctions et créez des modules de code réutilisables.

Support de la souris complètement intégré pour la selection des options de menu, les champs de saisie, et pour déplacer et redimensionner une fenêtre.

Portabilité. Le code de C-Scape est indépendant du hardware. Supporte DOS, OS/2, UNIX, AIX, VMS, QNX, XENIX... Détecte automatiquement Hercules, CGA, EGA, VGA, et les modes 43 et 50 lignes.

Edition de texte. Comprend un éditeur de texte avec contrôle de la césure, fonctions rechercher/remplacer, et manipulation des blocs de texte.

Flexibilité des champs. Champs masqués, protegés, marqués, avec label, obligatoires, avec ou sans echo. Validation complète des données. Champs de type heure, date, montant, liste déroulante, et autres fonctions de haut niveau. Créez votre type de champ.

Fenêtres. Encadrées, pop-up, superposées, eclatées, ascenceur, défilement horizontal; taille et nombre des fenêtres limités par la memoire RAM uniquement.

Menus. Déroulants, hiérarchiques, pop-up, style Lotus 1-2-3, ou créez le vôtre.

Système d'aide contextuel. Prompts et messages d'aide attachés à un écran ou à un champ. Messages à références croisées pour créer un système d'aide en ligne de style hypertexte.

Gestion des écrans. Ecrans stockés dans des fichiers ou linkés avec l'application.

Génération de code. Créez n'importe quel écran avec l'outil interactif Look & Feel, testez-le, puis convertissez-le automatiquement en code C.

Essayez C-Scape ! C-Scape est *puissant, flexible, portable*, d'apprentissage et d'utilisation facile. C-Scape est livré en standard avec la totalité du code source de la bibliothèque et l'outil de création d'écrans Look & Feel. La documentation est claire, extensive et detaillée. Avec C-scape, pas de module résident et pas de royalties sur les applications diffusées.

C-Scape est un produit d'**Oakland Group**. Smartline assure le support technique de C-scape par téléphone et par fax ainsi que des sessions de formation sur le produit.

Appelez-nous au (1) 46 34 64 19
ou envoyez le bon de commande ci-dessous par fax ou par courrier. Après le plaisir de C-Scape, le développement ne sera jamais plus ce qu'il était.

C-scape et Look & Feel **version 3.2** : DOS, OS/2 (Borland, Microsoft, Zortech): 4500FHT; UNIX: 13500FHT; Autres OS: nous consulter. Cartes de crédit acceptées.

smartline

30, rue Saint André des Arts
75006 Paris France
Fax: (1) 40 46 85 89 ***Tél: (1) 46 34 64 19***

Oakland Group, Inc 675 Massachusetts Ave, Cambridge, MA 02139 USA.Tél: 617-491-7311. Les produits d'Oakland sont distribués en Australie, Belgique, Danemark, Autriche, Allemagne, Suisse, Norvège, Suède, Angleterre, France. La photographie représente un programme C-Scape combinant une saisie de données avec des images vidéo chargées depuis un fichier .PCX. C-Scape et Look & Feel sont des marques déposées par Oakland Group, Inc.; les autres marques citées appartiennent à leurs propriétaires respectifs. Les fonctionnalités, prix et conditions sont sujets à modification. Documentation en anglais. C-scape (seascape): paysage de mer.

Service lecteurs n° 8 (dernière page)

✂

❑ Je désire recevoir une documentation sur C-scape

❑ Je désire recevoir le dossier d'évaluation comprenant les deux disquettes de démonstration C-scape et Look & Feel. Ci-joint un chèque de 120 F TTC. Une facture sera jointe à l'envoi.

nom prénom _____ société _____ tél _____

adresse _____

l'unité centrale. Les processeurs parallèles VLSI (Very Large Scale Integration) assurent une puissance de traitement variant de 150 à 300 MIPS, selon la configuration. Autres critères, les cartes Vitec gèrent une image de 32 Ko x 32 Ko, codée sur 24 bits, et génèrent des images pixel par pixel en 3D. Ainsi, il est possible d'associer une photographie, prise par un satellite, sur un relevé topographique en fil de fer. L'utilisateur pourra selon ses besoins implanter jusqu'à 264 Mo de Ram et 1,9 Go de disque dur.

Vitec

• Fabriqué par Visual Information Technologies • Distribué par Théta Scan •Carte dédiée au calcul et au traitement d'image • Format 8U • Processeurs parallèles VLSI • Puissance de 150 à 300 MIPS • Plate-forme Unix • Environnement C et XWindows • Systèmes SUN, HP, SGI, DEC et IBM •

Coté logiciel, le produit s'exploite à travers trois principaux modules en intéraction, dotés d'une interface graphique et conviviale. PICES est un environnement de calcul et de programmation autorisant l'accès direct au matériel, et supportant la manipulation de fichiers de texte et d'image. SCRIPT, langage L4G basé sur l'Anglais courant, permet la création de prototypes et le teste d'algorithmes par de simples commandes.

Enfin, l'option ADS + génère des micro-codes pour les processeurs VLSI. Un utilitaire joue le rôle d'un préprocesseur transforme les programmes SCRIPT en langage C. Le distributeur propose une dernière option la solution NICS, notament pour la mise en place de ces calculateurs en réseau. Vu le coût d'achat de ces macines cette solution apparait intéressante à partager.

Accélérer AutoCAD et Mathematica

Intel commercialise un nouveau coprocesseur destiné à accélérer la vitesse des logiciels Autocad, AutoShade, 3D studio et Mathématica sur des machines à base de microproceseurs Intel 386 DX.

Ce coprocesseur, appelé RapidCAD, permet de faire tourner les logiciels de CAO jusqu'à 70% plus vite qu'une configuration 386 DX couplée à un coprocesseur mathématique Intel 387 DX. Ainsi, grâce à ses capacités de calcul en virgule flottante, la fonction "Hide" d'Autocad tourne 46% plus rapidement.

Le jeu de circuits RapidCAD est constitué de deux composants. Le RapidCAD-1 est un circuit de 132 broches combinant un microprocesseur 386 et un coprocesseur arithmétique 387. Il se met à la place du 386 d'origine. Le RapidCAD-2 est un circuit de 68 broches qui s'enfiche en lieu et place du coprocesseur mathématique. Il intègre l'interface logique pour le traitement des exceptions pour les machines compatibles PC.

RapidCAD

• Conçu et distribué par Intel • Coprocesseur arithmétique pour machine de type 386DX • Composé de deux circuits intégrés • Augmentation de vitesse jusqu'à 70 % pour les produits de CAO par rapport à une machine de type 386DX équipé d'un classique 387 •

LES SEMINAIRES
ETAT DE L'ART

Micro & Communications

15-17 JUIN OU 21-23 SEPTEMBRE 1992
Micros: la synthèse 1992
Nouveaux choix techniques et stratégiques, perspectives à 3 ans.

18-19 JUIN OU 26-27 NOVEMBRE 1992
Applications Client-Serveur
Principes, techniques, outils disponibles, démarche pratique de mise en œuvre.

10-12 JUIN OU 21-23 SEPTEMBRE 1992
Unix et Systèmes Ouverts
Les connaissances indispensables : la synthèse en 3 jours.

18-19 JUIN OU 26-27 NOVEMBRE 1992
Connexion de Micros
Connexion aux mainframes, réseaux locaux de micros : les solutions actuelles.

22-24 JUIN OU 16-18 NOVEMBRE 1992
Bureautique
Choix techniques et stratégiques pour les 5 prochaines années.

15-16 SEPTEMBRE 1992
PC Management
Gestion des parcs micros : organisation, méthodes, outils.

22-24 JUIN OU 23-25 NOVEMBRE 1992
Réseaux Locaux
Etat de l'art et perspectives, architectures et alternatives techniques, panorama de l'offre commerciale.

24-25 SEPTEMBRE 1992
Sécurité Micro
Méthodes, outils, démarche pratique pour mettre en œuvre un plan de sécurité efficace.

Programmes détaillés et inscriptions : Véronique Groud, CAP SESA INSTITUT - 207, rue de Bercy, 75012 Paris Tél. : 44.74.24.10

CAP SESA INSTITUT
L'Etat de l'Art

Service lecteurs n° 9 (dernière page)

Re-sources

Samsung Graphic Station

Samsung se lance dans un secteur de pointe. Ce constructeur, traditionnellement spécialisé dans la micro-informatique, devrait prochainement proposer deux nouvelles stations graphiques, prêtes à l'emploi, livrées avec X Windows, version X11R4. Le premier modèle, SGS 19M, est construit autour d'un processeur AMD 29 000, cadencé à 16 MHz. Cette station est dotée en standard de 2 Mo de DRAM extensible à 18 Mo, de 768 Ko de NVRAM, et de 256 Ko de VRAM. Le tout de type SIMM.

L'écran monochrome, avec 19 pouces de diagonale, offre une résolution de 1024 X 1280 points. Le taux de rafraîchissement est de 66 Hz en mode non-entrelacé. Côté reseau, la SGS 19M supporte Ethernet Stats, Protocol Stats, Ethernet activity LED, ARP Table, ainsi que Routing Stats. Elle est d'ailleurs livrée avec une interface Ethernet AUI 15 Pin.

Le deuxième modèle est encore plus puissant. En effet, la SGS 17C contient un processeur AMD 29000, d'une cadence supérieure : 20 MHz. En standard, le produit est livré avec 2Mo de DRAM, extensibles à 10, 768 Ko de PROM, 2 Ko de NVRAM et 1 Mo de VRAM.

Samsung Graphic

• Fabriqué et distribué par Samsung • Modèles à base d'AMD 29000 cadencés à 16 ou 20 MHz • Livrés avec Windows X11R4 • Ecran 17 pouces haute résolution couleur ou monochrome • Interface et support réseau Ethernet •

Les performances graphiques divergent quelque peu : l'écran de 17 pouces couleur affiche une résolution de 1024 x 768 points en 256 couleurs sélectionnables parmi une palette de 16 millions. Il est doté d'une fréquence de rafraîchissement de 60 Hz en mode non-entrelacé. La SGS 17 C supporte les mêmes réseaux que la SGS 19M.

SQLBaseServer 5.0

Gupta Technologie lance sa base de donnée relationnelle SQLBase Server 5.0 en environnement Net Ware 3.11 de Novell. SQLBase fonctionne en mode natif sous NetWare en tant que

SQLBAseServer 5.0

• Edité et distribué par Gupta • Base de donnée relationnelle en environnement NetWare 3.11 de Novell • Solutions d'architecture client serveur • Niveau de performances élevées •

NLM (NetWare loadable Module). Les clients de Novell et Gupta ont ainsi la possibilité de bâtir des solutions d'architecture client-server apportant un nouveau niveau de performances : SQLBase Server pour NetWare atteint 67 transactions par seconde. SQLBase est la base de données relationnelle de Gupta. Récemment lancée, la version 5.0 offre un support optimisé des applications Windows. Elle est également disponible pour les plates-formes DOS, OS/2 et Unix. Cette nouvelle version offre notamment l'évolution des performances en fonction des plates-formes. Combinée avec SQLNetwork de Gupta, elle permet d'accéder à d'autres bases de données comme DB2, Oracle, Sybase...

Nouvelle version de X.Desktop sous Unix et VMS

Integral Software distribue la nouvelle version de X.Desktop 3.0 qui est une application permettant de créer un environnement de travail visuel et intuitif sous Unix et VMS. Programmes, fichiers et utilitaires sont représentés sous formes d'icônes. Ces icônes lancent des applications, gèrent les fichiers et donnent accès aux ressources et périphériques par les principes intuitifs apportés par les environnements graphiques. X.desktop est multi-plateforme : Sun, HP, DEC, IBM, Mips, DG, Bull... L'homogénéité obtenue en matière de présentation réduit les coûts de formation et de support et Unix et VMS accessibles aux utilisateurs non techniques. De même, l'environnement de travail visuel et convivial bénéficie également aux utilisateurs expérimentés en leur assurant un confort de travail et un gain de productivité importants. X.Desxtop est entièrement configurable pour répondre aux besoins de chaque utisateur, sans programmation complexe. Un éditeur d'icônes et une aide hypertexte en lignes sont disponibles.

X.desktop 3.0

• Edité et distribué par Integral SoftWare • Environnement graphique intuitif sous Unix et VMS - Améliore la convivialité d'Unix • Environnement reconfigurable • Editeur d'icône • Aide hypertexte en ligne •

Next sur PC

Nextstep, le logiciel système orienté objet de Next est disponible pour les ordinateurs équipés d'un microprocesseur 486. Si l'on se réfère aux chiffres fournis par Intel, la base installée de machines 486 peut être évaluée à environ deux millions.

Nextstep486 est identique au logiciel système orienté objet proposé sur les ordinateurs Next et inclut l'environnement de développement de Next ainsi que des applications telles que Nextmail, Digital Librarian et Digital Webster.

Le logiciel fonctionne bilatéralement sur les deux machines, ce qui rend possible la création de réseaux

NextStep486

• Nextstep pour machine à base de 486 • Disponible en version développeur et en version utilisateur • Transposition Next vers PC simple et fiable • Configuration matérielle requise importante •

utilisant à la fois des ordinateurs Next sous Nextstep486 et la communication avec des PCs sous DOS et Windows, des Macintosh et des ordinateurs sous Unix. La transposition des applications Next actuelles sur Nextstep486 s'avère être une opération simple et relativement fiable.

Nextstep486 nécessite de 8 à 16 Mo de mémoire, 120 Mo sur disque pour la version utilisateur et 400 Mo pour la version développeur.

L'installation peut être réalisée à partir d'un CD-ROM.

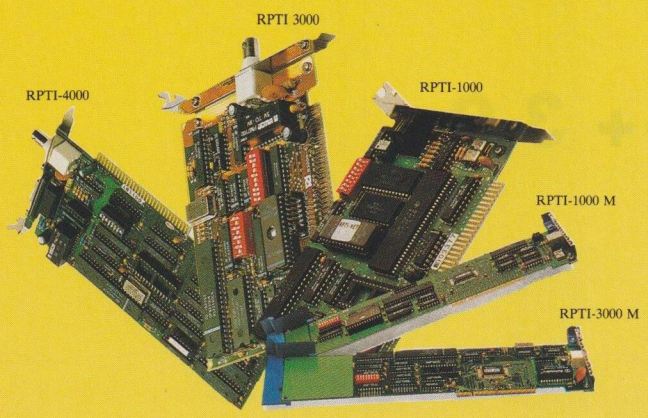
Fabian Pallini
Matthieu Thfoin

DISTRIBUTEUR AGRÉÉ NOVELL

QUALITÉ - SERVICE - DISPONIBILITÉ

Votre réseau local au prix direct importateur

Satisfait ou remboursé sous 15 jours



RÉSEAU RPTI-NET pour PC XT/AT/386 & 486

- Simplicité - Souplesse - Efficacité - Installation facile.
- Livré en Kit complet : Carte adaptateur + Logiciel N.O.S. + Connecteur.
- Logiciel N.O.S. : Serveur non dédié. Partage des ressources (disque, lecteur, imprimante, logiciel). message interactif.
- **NOUVEAUTÉ :** Le gestionnaire N.O.S. V.1.44 occupe 60 ko et se charge en mémoire haute. N.O.S. supporte DOS 5 WINDOWS 3, DR DOS.

Coût/Station

ETHERNET - Câble coaxial souple RG-58 A/U
R3000 : Carte 8 bits, 10 Mb/s + logiciel N.O.S. ..... **1490 F**
R4000 : Carte 16 bits, 10 Mb/s + logiciel N.O.S. ..... **1590 F**

ETHERNET 10 Base T - Paire torsadée RJ-45
R3000T : Carte 8 bits, 10 Mb/s + logiciel N.O.S. ..... **1590 F**
R4000T : Carte 16 bits, 10 Mb/s + logiciel N.O.S. ..... **1690 F**

ADAPTATEUR RÉSEAU FONCTIONNANT SOUS NOVELL NETWARE & NETWARE LITE

ETHERNET - Câble coaxial RG 58-58 A/U, vitesse 10 Mb/s
E1000 : Carte 8 bits, compatible NE-1000 ..... **890 F**
E2000 : Carte 16 bits, compatible NE-2000 ..... **990 F**
EP-001 : Pocket adaptateur pour les portables ..... 1950 F
ER-002 : Répéteur Ethernet, 2 ports BNC ..... **2500 F**

TRANSCEIVER
EH-001 : AUI - RJ-45 Transceiver ..... **450 F**
EH 002 : BNC - AUI Transceiver ..... **450 F**

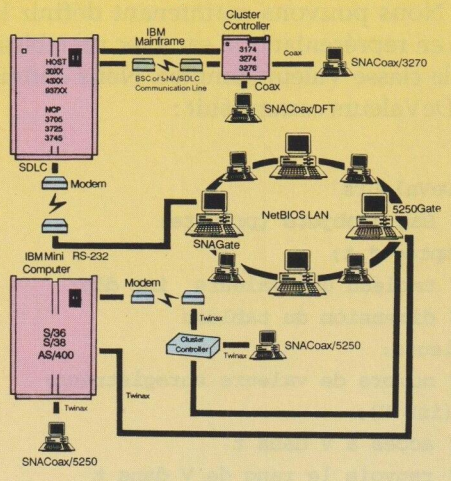
PROMOTION :
1 carte E1000 + 1 connecteur + NETWARE LITE ... **1700 F**

ETHERNET 10 Baes T, câble RJ-45, vitesse 10 Mb/s
E1000T : Carte 8 bits, compatible NE-1000 ..... **1090 F**
E2000T : Carte 16 bits, compatible NE-2000 ..... **1190 F**
EP-001T : Pocket adaptateur pour les portables ..... 2150 F

CONCENTRATEUR 10 BASE T, (HUB)
EH-006A : 6 ports RJ-45, 1 port BNC ..... 2850 F
EH-006B : 6 ports RJ-45, 1 port AUI (15 broches) .... **2850 F**
EH 012 : 12 ports RJ-45, 1 port BNC, 1 port AUI **5350**

NOVELL NETWARE - LAN MANAGER - IBM PC LAN - TCP/IP - UNIX - PASSERELLE 3270 & 5250 - EMULATION 3270 & 5250
Driver : Netware V2.1x, V3.0, V3.1, 3 com 3 + Share, Microsoft LAN MANAGER V1.x, V2.0, FTP TCP/IP, DEC DECnet, DOS/PCSA V2.1, V3.0, SUN PC-NFS. Nous consulter pour plus de renseignements.

"Open SNA Series"



Adaptateur réseau Ethernet Coax., Bus ISA + Driver
PB4111 : Open Ether/N1C, 8 bits ..... **1250 F**
PB4112 : Open Ether/N2C, 16 bits ..... **1450 F**
PB4113 : Open Ether/WDC, 8 bits ..... **1350 F**

Adaptateur réseau Ethernet 10 Base T, Bus ISA + Driver
PB4101 : Open Ether/N1, 8 bits ..... **1350 F**
PB4102 : Open Ether/N2, 16 bits ..... **1550 F**
PB4103 : Open Ether/WD, 8 bits ..... **1400 F**

Adaptateur Token Ring, Bus ISA + Driver
PB1101 : Open Token/16, 4 Mb/s ..... **2050 F**
PB1104 : Open Token/16A, 16/4 Mb/s ..... **3350 F**

PASSERELLE SNA 3270/5250 pour réseau NETBIOS
PB2400 : SNA 3270 LAN GateWay, 16 sessions ..... **9050 F**
PB2401 : SNA 3270 LAN GateWay, 32 sessions ..... **15500 F**
PB2403 : SNA 5250 LAN GateWay (S/36/38, AS400) ..... **11900 F**

Emulation SNA 3270
PB2406 : Carte + (Prise BNC) + logiciel ..... **4950 F**

Emulation SNA 5250 (AS400)
Carte + (Prise TWINAX) + logiciel ..... **4950 F**

Service lecteurs n° 10 (dernière page)

Service : Nous installons des réseaux du plus simple au plus complexe (Arcnet, Ethernet, 10 Base T, Token Ring). NOVELL NETWARE - UNIX - TCP/IP - Télématique sont nos domaines de compétences. **Etude :** Conseil, choix de l'architecture, extension du réseau existant. **Développement :** Interface de communication, applications fonctionnant sur Micro (monoposte - réseaux), sur AS400 et sur des gros systèmes (IBM 3090 - DPS7000).
SHOW-ROOM : Démonstration sur rendez-vous et sur mesure. Vous pouvez tester nos matériels et nos compétences.

Bon à renvoyer à **P.B. Net Informatique** - 15, rue de Reuilly - 75012 Paris. Tél. : 43 56 01 64 - Fax : 43 56 68 33

NOM : _____ Société : _____
Adresse : _____
Téléphone : _____ Télécopie : _____
☐ Je désire prendre rendez-vous avec un ingénieur technico-commercial
☐ J'ai l'intention d'acheter ☐ immédiatement ☐ dans le mois ☐ dans 3 mois

PB NET INFORMATIQUE

Nos prix s'entendent Hors Taxe, T.V.A. à 18,6 %, matériel garanti 1 an pièces et main-d'œuvre retour atelier.

IBM, NOVELL sont des marques déposées.

LANGAGE

LES «TEMPLATES» DE C++ avec Borland C++ 3.0

GILLES CLAVEL

Pourquoi les templates? A la différence d'un langage comme Smalltalk, C++ est un langage objet compilé, ce qui garantit de bonnes performances à l'exécution. En revanche, le polymorphisme est plus difficile à réaliser en C++. Cette caractéristique fondamentale des langages objets permet à des objets de classes différentes de réagir de manière appropriée à des messages semblables. Le polymorphisme était jusqu'ici obtenu partiellement en C++, avec l'utilisation des fonctions virtuelles. Mais, jusqu'à la version 2.1 d'AT&T, il manquait le paramétrage des types. Par exemple, pour obtenir en C++ 2.0, le polymorphisme d'une classe Tableau et ranger, dans un objet de cette classe, aussi bien des entiers que des réels ou des chaînes de caractères, il fallait définir autant de classes Tableau que de types de valeurs à enregistrer dans un tableau 1. Nous allons étudier, à travers un exemple, l'utilisation des templates, puis nous élargirons le sujet pour présenter l'environnement de développement de la nouvelle version de Borland C++, disponible depuis le début de l'année.

Exemple d'application

Supposons que nous voulions construire une classe dont les objets permettent d'enregistrer des valeurs non forcément distinctes. Une telle classe existe dans de nombreux environnements objets et est inspirée de la classe Bag de Smalltalk.

```
class ValeurCompte
        // permet de représenter des valeurs
        avec répétition
{private:
 int valeur;     // valeur répertoriée
 int ctr;        // compteur de répétitions
public:          // accès aux membres privés
 int Valeur() {return valeur;}
 int Ctr() {return ctr;}
        // constructeur pour la première
 occurrence de V
 ValeurComptee(int V)
   {valeur=V; ctr=1;}
        // constructeur par défaut
ValeurComptee() {};
        // ajouter une répétition
void UneDePlus() {ctr++;}
};
        // class ValeurComptee
```

Une instance de ValeurComptee est l'association d'un entier (valeur) et d'un compteur d'occurrences de cet entier (ctr). La classe ValeurComptee fournit deux constructeurs dont l'un construit l'objet correspondant à l'enregistrement d'une première occurrence de la valeur V. Nous pouvons maintenant définir la classe souhaitée, en représentant un sac avec un tableau d'instances de la classe ValeurComptee. Nous définissons la classe SacDeValeurs comme suit :

```
class SacDeValeurs
        // Sac d'objets {private:
 ValeurComptee * t;
        // tableau des valeurs  int dim;
        // dimension du tableau
 int nbValeurs;
        // nombre de valeurs enregistrées;
 int rang(int V);
        // accès à V dans t
        // renvoie le rang de V dans t
        ou une indication d'absence
 public:
        // constructeur d'un sac vide, de la capa-
cité C
 SacDeValeurs(int C)
{t = new ValeurComptee[C];  dim = C; nbValeurs = 0;}
        // destructeur
 ~SacDeValeurs() {delete [dim] t;}
 void AjouteValeur(int V);
        // ajoute V dans le sac
```

Borland C++ 3.0 est à la fois un environnement de développement pour DOS et pour Windows. Outre une interface plus riche et plus conviviale, il offre de nouvelles fonctionnalités, telles que les templates apparus avec la version 2.1 du C++ AT&T, avec lequel il est totalement compatible. Les templates permettent le paramétrage des types et simplifient considérablement le travail des concepteurs d'une bibliothèque de classes.

```
        // affiche le contenu du sac à l'écran
void AfficheToi(const char * Titre);
};
        // class SacDeValeurs
```

Une instance de la classe SacDeValeurs est représentée par un tableau t d'objets de la classe ValeurComptee, dont la dimension est donnée par le membre dim. Le membre nbValeurs indique le nombre (< dim) de valeurs distinctes du sac. Le constructeur SacDeValeurs(int C) permet de créer un sac, vide au départ et pouvant accueillir C valeurs distinctes. Nous ne détaillerons pas ici l'implémentation des fonctions membres qui ne sont pas complètement précisées dans la définition de la classe SacDeValeurs. Si nous supposons cependant que la définition de la classe est complète, on peut compiler et exécuter une séquence telle que :

```
SacDeValeurs MonSac(5);
MonSac.AjouteValeur(4);
MonSac.AjouteValeur(2);
MonSac.AjouteValeur(3);
MonSac.AjouteValeur(4);
MonSac.AfficheToi("MonSac");
```

qui produira, par exemple, l'affichage :

```
Contenu du SacDeValeurs MonSac
4 présent 2 fois
2 présent 1 fois
3 présent 1 fois
```

Définir une classe paramétrée

Si la classe SacDeValeurs nous convient, comment pouvons-nous l'utiliser pour représenter des sacs de valeurs réelles ou de valeurs d'une classe que nous aurons nous-mêmes définie?

Avant la version 2.1, il n'y avait pas d'autre solution que celle de définir, pour chaque type XXX de valeurs à gérer, une classe SacDeValeursXXX, avec un nom différent pour chacune de ces classes.

Avec les templates, nous pouvons maintenant définir une seule classe paramétrée et laisser le compilateur faire le travail à notre place :

```
template <class Objet> class SacDeValeurs
        // Sac d'objets
{private:
 class ValeurComptee
        // Objet Répété
{private:
 Objet valeur;
        // valeur répertoriée
 int ctr;
        // nombre d'occurrences de cette valeur
 public:
 Objet Valeur() {return valeur;}
 int Ctr() {return ctr;}
 ValeurComptee() {};
 ValeurComptee(Objet V);
 void UneDePlus () {ctr++;}   };
        // class ValeurComptee  ValeurComptee *
t;      // tableau des valeurs  int dim;
        // dimension du tableau
 int nbValeurs;
        // nombre de valeurs enregistrées;
 int rang (Objet V); public:
 SacDeValeurs(int Capacite);
 ~SacDeValeurs() {delete [] t;}
 void AjouteValeur(Objet V);
 void AfficheToi(const char * Titre); };
        // class SacDeValeurs
```

Par rapport à la version précédente, la spécification :
`template <class Objet>`
transforme la classe SacDeValeurs en un modèle de classe (class template) qui décrit comment le compila-

teur doit construire les classes paramétrées avec l'argument Objet. On notera l'emboîtement (nesting) de la classe ValeurComptee dans la classe SacDeValeurs, qui fait que ValeurComptee est également un modèle de classe, avec le même argument Objet.

Avec la version AT&T 2.1, la portée d'une classe emboîtée est limitée à la classe emboîtante. Ainsi, en dehors de la définition de la classe SacDeValeurs, la classe ValeurComptee devra être désignée par SacDeValeurs::ValeurComptee. Comment utiliser un modèle de classe? Il suffit de préciser la «valeur» du template, dans la déclaration des objets souhaités :

```
class CodeVille { ... }
      // un objet CodeVille associe un code
postal et la ville correspondante
SacDeValeurs<int>::ValeurComptee A(11);
SacDeValeurs <int> Sac1(5);
SacDeValeurs <float> Sac2(10);
SacDeValeurs <CodeVille> Sac3(10);
```

On notera enfin que le mécanisme des templates peut être utilisé pour définir des fonctions génériques dont un ou plusieurs arguments doivent être de type variable. Supposons que la fonction AfficheToi de la classe SacDeValeurs soit définie par :

```
template <class Objet>
void SacDeValeurs<Objet>::AfficheToi
    (const char * Titre)
  {cout << "Contenu du SacDeValeurs "
    << Titre << "\n";
  for (int K=0; K<nbValeurs; K++)
    {Affiche(t[K].Valeur());
    // t[K].Valeur est du type Objet
    cout <<" présent "
         <<t[K].Ctr() <<" fois\n";}
} // SacDeValeurs::AfficheToi
```

Dans cette fonction, l'affichage d'une valeur du type Objet est réalisé par appel à la fonction Affiche qui, si Objet correspond à int ou à float, sera définie par :

```
void Affiche(int E) {cout << E;} void
Affiche(float R) {cout << R;}
```

Mais si Objet est une classe dont les valeurs ne sont pas acceptées par l'opérateur << des flux C++, il nous faudra soit redéfinir cet opérateur, ce qui implique un travail complexe et dangereux, soit, tout simplement, définir le modèle de fonction (function template) :

```
template <class Valeur>
void Affiche(Valeur V) {V.AfficheToi();}
```

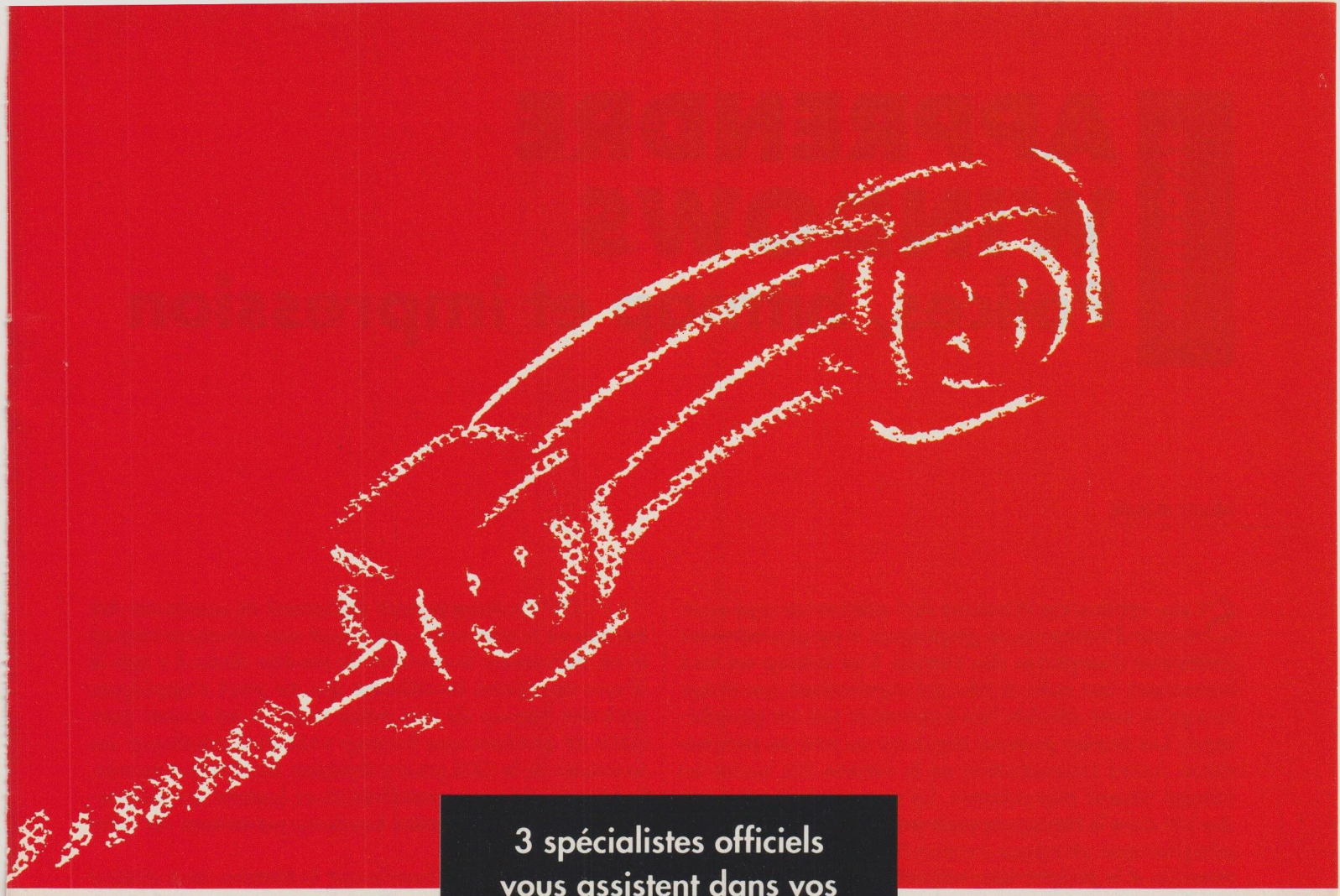
en s'assurant, bien entendu, que chaque classe susceptible de correspondre à Objet définisse bien la fonction-membre AfficheToi() adéquate.

Borland C++ 3.0

Borland C++ Application Frameworks est un environnement MS-DOS/Windows de développement qui s'installe simplement et s'utilise avec MS-DOS, comme avec Windows. Le produit comporte :

- Les langages C (norme Ansi) et C++. Les compilateurs C++ fournis représentent, en janvier 1992, la première implémentation complète de la version AT&T 2.1.
- Deux environnements de développement C++ sont proposés pour développer sous DOS ou Windows. Borland C++ permet de développer des applications DOS ou Windows, et son environnement de développement intégré peut être utilisé sous DOS ou sous Windows en fenêtre DOS. Turbo C++ est un environnement de développement complètement intégré à Windows et qui constitue en quelque sorte la réponse de Borland au Quick C/Windows de Microsoft avec, en plus, les mécanismes objets de C++ et la librairie de classes ObjectWindows. Bien entendu, Borland C++ peut aussi utiliser ObjectWindows.
- L'intégration à l'environnement Windows vous permet également, avec le simple basculement d'une option de compilation, de transformer une application dos qui utilise des fonctions standard DOS d'entrée-sortie clavier-écran (printf, scanf ...) en une application qui tournera dans une fenêtre Windows.
- L'utilisation systématique du mode protégé (standard DPMI) permet l'emploi de la mémoire étendue pour la compilation et l'édition de liens avec des performances excellentes. L'option de précompilation des en-têtes augmente encore ces performances, ce qui est particulièrement appréciable avec les en-têtes pour Windows qui ne sont pas loin de contenir 10 000 lignes de code.
- Bien sûr, l'environnement de développement propose un débogueur sous Windows (icône TDW), et l'environnement Turbo C++ offre un explorateur graphique de classes (class browser) bien conçu et agréable à utiliser. Pour la mise au point des applications Windows, un nouvel utilitaire permet de suivre l'enchaînement des messages traités (icône Winsight) et, évidemment, une aide en ligne complète.

Beaucoup de choses donc avec ce nouveau produit. Mais 45 Mo d'espace disque sont nécessaires pour l'installer. C'est la rançon du confort et des performances! ❐



3 spécialistes officiels vous assistent dans vos développements Novell.

Développeurs, décrochez le téléphone rouge.

Vous n'êtes jamais seul avec un outil de développement Novell.
Dans le cadre de son "Professional Developer's Program", Novell vous garantit l'assistance totale de trois Importateurs Officiels. Vous pouvez tout leur demander.
Assistance logistique, conseil et équipement rapide direct pour développer sous Novell.
Assistance par la formation qualifiée des développeurs.

Notez leurs noms à l'encre rouge.

Alfratronic
ZI de Courtabœuf
7 av. du Canada - B.P. 310
91958 Les Ulis Cédex
Tél. 69 86 75 00

Interquad
64-66, rue Escudier
92100 Boulogne
Tél. 46 84 05 15

Omnilogic
11, rue de Cambrai - Bât. 28
75019 Paris
Tél. 40 05 28 00

NOVELL

Le passé, le présent, l'avenir de l'informatique réseau

HARD communication

WINDOWS

APPRENDRE WINDOWS

Polices, bitmap et impression

MARC ADLER

Si une application Windows doit afficher un type quelconque d'informations graphiques, il est généralement préférable d'employer des bitmap. Une bitmap est tout bonnement une suite de bits qui représentent une image lorsqu'ils sont affichés (ou imprimés) par un périphérique. Les bitmap peuvent dépendre ou non du périphérique. Avant l'introduction de la version 3.0, toutes les bitmap étaient dépendantes. La représentation interne des bits dans la bitmap reposait sur les caractéristiques du périphérique physique employé pour l'affichage. Par exemple, si le périphérique utilisait des plans de bits pour représenter une image en couleurs, les bits de la bitmap devaient être organisés comme une série de plans. Windows 3.0, qui supporte toujours les bitmap dépendantes, a introduit les bitmap indépendantes du périphérique (DIB, pour «Device-Independant Bitmap»).

Ce type de bitmap décrit l'aspect d'une image plutôt que l'organisation physique de l'image dans le périphérique matériel. Le format DIB est employé par Windows pour toutes ses propres ressources bitmap, par exemple les icônes ou les curseurs de la souris. En fait, un utilitaire de conversion spécifique, fourni avec le SDK de Windows 3.0, permet de convertir les bitmap de Windows 2.x en bitmap au format DIB. Il est possible de créer des bitmap (ainsi que des icônes et des curseurs) avec le programme SDKPaint, puis de les sauvegarder dans un fichier ayant l'extension BMP. On peut placer une référence à une bitmap dans le fichier ressource d'une application, comme ceci :

```
Clock BITMAP clock.bmp
```

Quand le fichier ressource est compilé, une copie de la bitmap est placée dans le fichier EXE, avec toutes les autres ressources. L'image bitmap peut alors être utilisée par l'application à l'aide de la fonction LoadBitmap.

```
HBITMAP hBitmap;
hBitmap = LoadBitmap(hInstance,
(LPSTR) "Clock");
```

La GDI Windows permet également de construire des bitmap «à la demande», en créant une structure bitmap et en définissant le contenu de ses bits. Mais il s'agit là d'une opération fastidieuse qui ne peut s'appliquer qu'à des bitmap de très petite taille. Il est encore possible de générer une bitmap vide et d'employer les fonctions de la GDI pour y dessiner. En d'autres termes, la bitmap vide est considérée alors comme un périphérique de sortie. Plusieurs fonctions permettent de créer une bitmap vide.

Elles renvoient un handle sur la bitmap, tout comme la fonction LoadBitmap. Pour la conception de bitmap dépendant du périphérique, nous pouvons utiliser CreateBitmap et CreateBitmapIndirect. Ces fonctions ne devraient être employées que pour des bitmap monochromes, car les bitmap couleurs sont dépendantes des périphériques :

```
hBitmap = CreateBitmap(nWidth,  nHeight, nPlanes,
nBitCount, lpBits)
```

et

```
hBitmap = CreateBitmapIndirect  (BITMAP FAR
*lpBitmap)
```

Ainsi, une bitmap de 8 x 8 bits représentant une flèche s'écrit :

```
BYTE szArrow[] =
{
        0x18,
        // 00011000
        0x3C    // 00111100
        0x7E    // 01111110
        0xFF    // 11111111
        0x18,   // 00011000
        0x18,   // 00011000
        0x18,   // 00011000
        0x18,   // 00011000
```

Voici le dernier article de notre série «Apprendre Windows». Nous terminerons l'étude de la GDI Windows par les polices et les bitmap, puis nous examinerons les notions relatives à l'impression. Enfin, nous perfectionnerons notre application d'exemple pour qu'elle puisse profiter des avantages du gestionnaire d'impression de Windows.

```
};
HBITMAP  hBitmap;
hBitmap = CreateBitmap
              (8, 8, 1,  1, (LPSTR) szArrow);
```

Les bitmap monochromes sont (relativement) indépendantes du matériel. La plupart des périphériques de sortie supportent un concept simple : quand un bit est à un, le pixel correspondant est affiché avec la couleur d'avant-plan par défaut; quand un bit est à zéro, il est affiché avec la couleur d'arrière-plan par défaut.

Une façon plus «populaire» de créer une bitmap vide est l'emploi de CreateCompatibleBitmap. Cette fonction crée une bitmap dont le format interne est compatible avec un périphérique associé à un contexte de périphérique. Pour utiliser cette fonction, il faut obtenir un contexte de périphérique pour le périphérique de sortie courant :

```
HDC hDC, hMemDC; hDC = GetDC(hWnd);
```

Il faut alors récupérer un contexte de périphérique mémoire afin de pouvoir dessiner dans la bitmap. Ce contexte de périphérique mémoire doit être compatible avec le périphérique de sortie :

```
hMemDC = CreateCompatibleDC(hDC);
```

Ensuite, il faut créer une bitmap vide qui soit compatible avec le périphérique de sortie. Supposons que vous souhaitiez construire une bitmap de 32 x 32 pixels :

```
HBITMAP hBitmap;
hBitmap = CreateCompatibleBitmap
  (hDC, 32, 32);
```

Le premier argument de CreateCompatibleBitmap est le contexte de périphérique qui sera associé à la bitmap. Comme celle-ci sera éventuellement dessinée dans une fenêtre, elle doit avoir le même format que le périphérique sur lequel la fenêtre est affichée. Les deux arguments suivants sont la largeur et la hauteur de la bitmap.

Ces deux arguments sont exprimés en pixels, et non en bits (pour les bitmap monochromes, le nombre de bits équivaut au nombre de pixels, mais ces nombres sont différents lorsqu'il s'agit d'une bitmap couleurs). La bitmap vide doit ensuite être associée au contexte de périphérique mémoire.

Lorsque cela est accompli, notre bitmap devient la surface d'affichage de ce contexte mémoire. Tout ce qui est envoyé vers le contexte de périphérique mémoire se trouve reflété dans la bitmap.

```
HBITMAP hOldBitmap; hOldBitmap = SelectObject
(hMemDC, hBitmap);
```

Comme la bitmap n'a pas été initialisée, il faut d'abord peindre tous les pixels en blanc (ou en une autre couleur) avant de pouvoir dessiner.

```
PatBlt(hMemDC, 0, 0, 32, 32,  WHITENESS);
```

La fonction PatBlt passe une trame de bit sur le périphérique associé au contexte spécifié par le premier argument. Les arguments suivants de PatBlt représentent la position de départ, la largeur et la hauteur de la zone à peindre. Le dernier est un code ROP (abréviation de «Raster OPeration»). WHITENESS indique à PatBlt que des pixels blancs doivent être copiés sur le périphérique — en recouvrant tous les pixels qui se trouvent dans la zone.

Les autres codes ROP qui peuvent être employés par cette fonction sont BLACKNESS (pixels noirs), PATCOPY (copie la trame du pinceau courant), PATINVERT (combinaison du pinceau courant et de la bitmap de destination, à l'aide d'une opération XOR) et DSTINVERT (inversion de la bitmap de destination). Nous allons maintenant dessiner un grand X sur la bitmap vide :

FIGURE 1 : LA STRUCTURE BITMAPINFO

La structure BITMAPINFO définit complètement les dimensions et la couleur d'une DIB de Windows 3.0.

```
typedef struct tagBITMAPINFO {
        BITMAPINFOHEADER
        bmiHeader;
        RGBQUAD bmiColors[]1;
} BITMAPINFO;
```

La structure BITMAPINFO contient les champs suivants :

CHAMP	DESCRIPTION
bmiHeader	Spécifie une structure de données BITMAPINFOHEADER contenant des informations sur les dimensions et le format de couleur d'une DIB.
bmiColors	Spécifie un tableau de structures RGBQUAD définissant les couleurs de la bitmap.

Une DIB est constituée de deux parties distinctes : une structure BITMAPINFO décrivant les dimensions et les couleurs de la bitmap et un tableau d'octets définissant les pixels de la bitmap. Les bits du tableau sont contigus, mais chaque ligne de pixels doit être complétée par des zéros pour se terminer sur une limite LONG.
Des limites de segment peuvent se trouver n'importe où dans la bitmap. L'origine de la bitmap est le coin inférieur gauche. Le champ biBitCount de la structure BITMAPINFOHEADER détermine le nombre de bits employés pour définir un pixel et le nombre maximal de couleurs dans la bitmap. Ce champ peut prendre l'une des valeurs suivantes :

VALEUR	SIGNIFICATION
1	La bitmap est monochrome et le champ bmiColors doit contenir deux entrées. Chaque bit du tableau représente 1 pixel. Si le bit est à 0, le pixel est affiché avec la couleur de la première entrée de la table bmiColors; si le bit est à 1, le pixel prend la couleur de la seconde entrée de la table.
4	La bitmap a un maximum de 16 couleurs, et le champ bmiColors contient jusqu'à 16 entrées. Chaque pixel est représenté par un indice de 4 bits dans la table des couleurs. Par exemple, si le premier octet de la bitmap est 1FH, il représente deux pixels. Le premier pixel contient la couleur de la seconde entrée de la table, et le second pixel contient la couleur de la 16e entrée de la table.
8	La bitmap a un maximum de 256 couleurs, et le champ bmiColors contient jusqu'à 256 entrées. Dans ce cas, chaque octet du tableau de la bitmap représente un pixel.
24	La bitmap a un maximum de 224 couleurs. Le champ miColors a pour valeur NULL, et chaque groupe de trois octets dans le tableau de la bitmap représente respectivement les intensités relatives de rouge, vert et bleu d'un pixel.

Le champ biClrUsed de la structure BITMAPINFOHEADER spécifie le nombre d'indices de couleur dans la table des couleurs utilisée par la bitmap. Si le champ biClrUsed est à 0, la bitmap utilise le nombre maximal de couleurs correspondant à la valeur du champ biBitCount.
Les couleurs de la table bmiColors doivent être placées par ordre d'importance. Néanmoins, pour les fonctions utilisant les DIB, le champ bmiColors peut être un tableau d'entiers non signés sur 16 bits, spécifiant chacun un indice dans la palette logique courante au lieu d'une valeur RGB explicite.
Dans ce cas, une application qui utilise la bitmap doit appeler les fonctions DIB en donnant au paramètre wUsage la valeur DIB_PAL_COLORS.
Le champ bmiColors ne doit pas contenir d'indices de palette si la bitmap doit être stockée dans un fichier ou passée à une autre application.
A moins d'être utilisée exclusivement par l'application, et sous son contrôle absolu, la table des couleurs ne doit contenir que des valeurs RGB explicites.

```
MoveTo(hMemC, 0, 0);
// deux lignes en diagonale
LineTo(hMemDC, 31, 31);
MoveTo(hMemC, 0, 31);
// pour desiner un X
LineTo(hMemDC, 31, 0);
```

Nous avons dessiné quelque chose sur la bitmap, mais celle-ci est en mémoire. Il faut donc transférer l'image vers le périphérique de sortie à l'aide de la fonction BitBlt. Tous les systèmes graphiques possèdent une fonction pour déplacer des bits dans la mémoire. Les bits de la source peuvent être combinés de diverses façons avec ceux de la destination.

Ces combinaisons sont contrôlées par les codes ROP que nous avons cités précédemment. BitBlt est la fonction Windows qui déplace une bitmap d'un périphérique vers un autre. Dans le cas qui nous occupe, le périphérique source est un bloc de mémoire et le périphérique destination est la fenêtre dans laquelle la bitmap doit être affichée. Le code du transfert est le suivant :

```
BitBlt(hDC, 0, 0, 32, 32,  hMemDC, 0, 0, SRCCOPY);
```

Les cinq premiers arguments de BitBlt représentent le handle du périphérique destination, ainsi que les coordonnées de départ et la taille de la cible. Les quatre arguments suivants sont le handle de la bitmap source, ses coordonnées de départ et le code ROP à employer. Après avoir affiché la bitmap, il faut accomplir quelques opérations de nettoyage. Il est nécessaire de détruire le contexte de périphérique mémoire et de libérer le contexte d'affichage de la fenêtre. Et il faut réassocier l'ancienne bitmap au contexte de périphérique mémoire avant de détruire celui-ci.

```
SelectObject(hMemDC,hOldBitmap);
DeleteDC(hMemDC);
ReleaseDC(hWnd, hDC);
```

En examinant les arguments de BitBlt, on remarque qu'une seule zone est spécifiée. Les bits situés dans cette zone rectangulaire sont transférés un par un de la source vers la destination. Windows permet encore de déplacer une zone rectangulaire de la bitmap source vers une zone de taille différente dans la bitmap cible. Il est possible d'afficher la bitmap de 32 x 32, stockée dans le contexte de périphérique mémoire, sur une zone de 256 x 256 du périphérique de destination. Pour cela, les bits de la bitmap source doivent être "élargis" avant d'être placés dans la zone cible. Cette déformation (agrandissement ou réduction) est accomplie par la fonction StretchBlt. Celle-ci ressemble beaucoup à BitBlt, mais elle reçoit deux arguments de plus : la hauteur et la largeur de la bitmap source à transférer. Voici comment copier une bitmap de 32 x 32 dans une zone de 256 x 256 :

```
StretchBlt(hDC, 0, 0, 256, 256,
                hMemDC, 0, 0, 32, 32, SRCCOPY);
```

Une partie de l'opération de StretchBlt est contrôlée par un membre de la structure de contexte de périphérique : le mode de déformation (ou mode de «stretching»). Les opérations d'agrandissement et de réduction doivent être accomplies avec le plus grand soin pour préserver l'aspect de la bitmap originale. Le mode de déformation permet à StretchBlt de déterminer quels pixels doivent être combinés quand une bitmap est réduite, et il peut être défini par une application à l'aide de la fonction SetStretchBltMode. Le mode de déformation peut recevoir trois valeurs, dont deux ne sont importantes que pour les bitmap monochromes. BLACKONWHITE, mode par défaut, effectue un ET logique entre les bits retenus et ceux qui sont supprimés. Cela donne aux pixels noirs une priorité sur les pixels blancs. Le mode WHITEONBLACK préserve les pixels blancs au détriment des noirs, par une opération OU logique. Si l'on travaille avec des bitmap couleurs, il est possible d'utiliser le mode COLORONCOLOR, qui supprime simplement les bits qui n'ont pas été retenus.

Les bitmap indépendantes

Windows 3.0 propose de nouvelles fonctions GDI pour manipuler les bitmap indépendantes du périphérique (DIB). Les noms de ces fonctions sont identifiés aisément car ils contiennent les lettres DIB. CreateDIBitmap et StretchDIBits sont les équivalents DIB de CreateBitmap et de StretchBlt. On peut obtenir et définir les bits dans une portion de bitmap grâce aux fonctions GetDIBits et SetDIBits. Pour afficher directement une DIB sur un périphérique de sortie, on appelle SetDIBitsToDevice.

Ces fonctions emploient, soit la structure BITMAPINFO, soit la structure BITMAPINFOHEADER (qui est en fait une partie de BITMAPINFO, *voir figures 1 et 2*). Comme on peut le voir, la structure BITMAPINFO contient deux composants : un en-tête donnant des informations sur le format de couleur et les dimensions de la bitmap et un tableau d'informations de couleurs. Ce tableau peut contenir les valeurs RGB effectives de la bitmap ou des valeurs d'index sur la palette courante.

Si l'on crée la fonction DisplayDIBitmap pour afficher une bitmap indépendante dans une fenêtre, plusieurs

étapes sont nécessaires. Cette fonction doit recevoir comme paramètres le handle de la fenêtre dans laquelle se fera l'affichage, un contexte de périphérique pour cette fenêtre, un handle global sur les bits qui définissent la bitmap, ainsi que la largeur et la hauteur de cette dernière.

Nous supposerons qu'il s'agit d'une bitmap 16 couleurs, ce qui signifie qu'il faut 4 bits pour représenter un pixel. DisplayDIBitmap appelle d'abord AllocateDefaultBitmap pour allouer et initialiser une structure BITMAPINFO. Cette structure reçoit les dimensions et l'information de format de couleur d'une bitmap 16 couleurs. Un handle sur la structure est renvoyé à DisplayDIBitmap, qui se charge de déréférencer la structure. Pour afficher la structure dans la fenêtre, il suffit d'appeler StretchDIBits ou SetDIBitsToDevice.

Cette dernière fonction n'accomplit aucune déformation de la bitmap; en cela, elle est comparable à BitBlt. Si la bitmap doit être affichée sur toute la surface de la fenêtre, il faut utiliser StretchDIBits.

Remarquons un point important : il n'est pas nécessaire ici de créer un contexte mémoire temporaire pour transférer la bitmap vers le périphérique de sortie *(voir figure 3)*.

FIGURE 2 : LA STRUCTURE BITMAPINFOHEADER

La structure BITMAPINFOHEADER contient des informations sur les dimensions et le format de couleur d'une DIB.

```
typedef struct tagBITMAPINFOHEADER{
        DWORD  biSize;
        DWORD  biWidth;
        DWORD  biHeight;
        DWORD  biPlanes;
        DWORD  biBitCount;
        DWORD  biCompression;
        DWORD  biSizeImage;
        DWORD  biXPelsPerMeter;
        DWORD  biYPelsPerMeter;
        DWORD  biClrUsed;
        DWORD  biClrImportant;
}  BITMAPINFOHEADER;
```

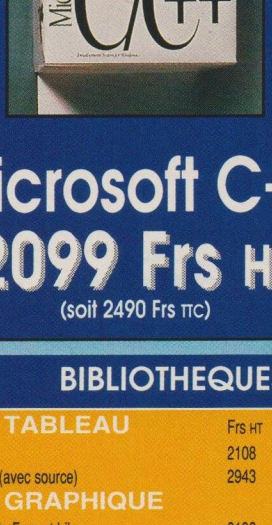
La structure BITMAPINFOHEADER possède les champs suivants :

Champ	Description
biSize	Spécifie le nombre d'octets requis par la structure BITMAPINFOHEADER.
biWidth	Spécifie la largeur de la bitmap en pixels.
biHight	Spécifie la hauteur de la bitmap en pixels.
biPlanes	Spécifie le nombre de plans pour le périphérique cible, et doit être mis à 1.
biBitCount	Spécifie le nombre de bits par pixel. Peut prendre pour valeur 1, 4, 8 ou 24.
biCompression	Spécifie le type de compression dans le cas d'une bitmap compressée. Il peut prendre une des valeurs suivantes :
Valeur	Signification
BI_RGB	Spécifie que la bitmap n'est pas compressée.
BI_RLE8	Spécifie un format RLE (Run-Length Encoded) pour les bitmaps ayant 8 bits par pixel. Le format de compression est un format sur 2 octets, composé d'un octet de comptage suivi par un octet contenant l'indice de couleur.
BI_RLE4	Spécifie un format RLE pour les bitmaps ayant 4 bits par pixel. Le format de compression est un format sur 2 octets, composé d'un octet de comptage suivi par deux indices de couleur sur un mot.
biSizeImage	Spécifie la taille de l'image, en octets.
biXPelsPerMeter	Spécifie la résolution horizontale, en pixels par mètre, du périphérique cible. Une application peut employer cette valeur pour sélectionner dans un groupe de ressources la bitmap qui correspond le mieux aux caractéristiques du périphérique.
biYPelsMeter	Spécifie la résolution verticale, en pixels par mètre, du périphérique cible.
biClrUsed	Spécifie le nombre d'indices de couleur dans la table de couleurs utilisée par la bitmap. Si cette valeur est 0, la bitmap utilise le nombre maximal de couleurs correspondant à la valeur du champ biBitCount. Si biClrUsed est différent de 0 et que biBitCount est inférieur à 24, il spécifie le nombre effectif de couleurs auxquelles pourront accéder le moteur graphique ou le pilote de périphérique. Si le champ biBitcount est à 24, biClrUsed représente la taille de la table de référence des couleurs qui est employée pour optimiser les performances des palettes de couleurs de Windows. Si la bitmap est compacte (c'est-à-dire si le tableau de la bitmap suit immédiatement l'en-tête de BITMAPINFO et si la bitmap est référencée par un seul pointeur), le champ biClrUsed doit être mis à 0 ou représenter la taille effective de la table des couleurs.
biClrImportant	Spécifie le nombre d'indices de couleur qui sont considérés comme importants pour l'affichage de la bitmap. Si cette valeur est 0, toutes les couleurs sont importantes.

40.18.10.70

!!! S.O.S. DEVELOPPEURS !!!

PROMO DU MOIS



Microsoft C++
2099 Frs HT
(soit 2490 Frs TTC)

COMPILATEURS

LANGAGE C ++	Frs HT	Frs TTC
MS C++ 7.0	2099	2490
Borland C++ 3.0 Fr	2317	2748
Borland C++ / Application Frame 3.0 Fr	3477	4124
Turbo C++ 2nde Edition Fr	647	767
Turbo C++ Windows Fr	867	1028
Turbo C++ & Turbo Vision Fr	1037	1230
Zorthec C++ Compiler	2487	2950
Zorthec C++ Development	3766	4466
BASIC		
MS Basic PDS 7.1 US	2594	3076
Quick Basic 4.5 Fr	644	763
LANGAGE C		
Quick C 2.5 Fr	644	763
Quick C / Windows Fr	1154	1369
MS C 6.0a	2198	2607
PASCAL		
Turbo Pascal 6.0 Fr	927	1099
Turbo Pascal 6.0 Professional Fr	1857	2200
Turbo Pascal / Windows Fr	1472	1746
MS Pascal PDS 4.0	2594	3076
Quick Pascal 1.0 Fr	644	763

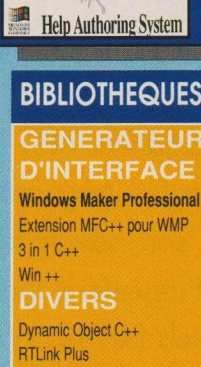
Support technique par les filiales françaises de chaque concepteur.

Windows Maker Professional

Développez vos applications Windows en cliquant sur votre souris, Windows Maker Pro génère directement tout le code C (documenté). Economisez des jours de développement et gagnez en maintenabilité. Testez le Look & Feel et effectuez les changements sans recompiler. Déjà Best Seller USA des développeurs C, la nouvelle extension MFC++ (module séparé) fait de Windows Maker Pro la référence des développeurs Microsoft C++ 7.0.

4900 Frs HT (soit 5811 Frs TTC)

!!! PROMOTION !!!
Windows Maker Pro & MFC++ & MS C++ 7.0
7390 Frs HT (soit 8765 Frs TTC)



Magic Fields

Toute la validation des champs de saisie est contrôlée (date, numérique...) par Magic Fields qui génère directement le code C. De plus, Magic Fields vous donne la possibilité d'offrir un look "NeXT" avec des reliefs à vos applications.

Nouveau !!!

RoboHELP

La génération des fichiers d'aide Windows en Hypertext n'a jamais été aussi simple et rapide. RoboHELP s'appuie sur Windows Word pour vous aider à réaliser vos fichiers Hypertext en cliquant sur la souris.

Nouveau !!!

BIBLIOTHEQUES / OUTILS SPECIAL WINDOWS

OBJET TABLEAU	Frs HT	Frs TTC
Soft Fields	2108	2500
Object Table C (avec source)	2943	3490
OBJET GRAPHIQUE		
HALO Image File Format Lib	2192	2600
Soft Chart	2951	3500
TIFF, GIF, PCX, Image : SDK	*nous consulter*	
OPTIMISATION		
MemShield 1.0	3356	3980
OptiMem (C & C++)	3024	3587
WindExe	2782	3300
ACCES AU SGBD		
Q+E DataBase Library Windows	2656	3150
QEVB (SPECIAL VISUAL BASIC)	1400	1660
Code Base 4.5	2867	3400
DataLib	3583	4250
COMMUNICATION		
Distinct TCP/IP 2.0a	*nous consulter*	
Chameleon TCP/IP 3.0	*nous consulter*	
TEST		
Microsoft Test	1850	2194
SQA Robot 1.1	*nous consulter*	
PORTAGE WINDOWS / DOS / OS/2		
MEWEL	2980	3535
WinPort	3200	3795
GENERATEUR D'INTERFACE		
Windows Maker Professional 3.10	4900	5811
3 in 1 C	1813	2150

VALIDATION DE SAISIES		
Magic Fields 1.0	**2800**	**3321**
Dialog Coder 3.0	4123	4890
EDITEUR / DEBUGGER		
ICE/W	*nous consulter*	
Codewrite	2437	2890
Periscope 32 Debugger Windows	2487	2950
DIVERS		
RoboHELP 1.0	**3800**	**4507**
ReportRight	*nous consulter*	
Drover's Toolbox	2487	2950
Kansman Pack	4115	4880
(avec source : table / statut / ribbon / toolbox)		
GENERATEUR D'APPLICATION		
Object Vision 2.0	3187	3780
Infiné/2 Windows	13406	15900
Spinnaker (Windows / Mac)	*nous consulter*	
Realizer	*nous consulter*	
SPECIAL VISUAL BASIC		
Visual Basic	1154	1369
Professional ToolKit for Visual Basic	1533	1818
Visual Basic Lib. & SQL SDK	2312	2742
QEVB	1400	1660
Visual Architect	1927	2286
VB Assist	1710	2028
DDE Tools	*nous consulter*	

BIBLIOTHEQUES / OUTILS SPECIAL C++

GENERATEUR D'INTERFACE	Frs HT	Frs TTC
Windows Maker Professional & MFC++	**5699**	**6759**
Extension MFC++ pour WMP	3377	4005
3 in 1 C++	1644	1950
Win ++	1644	1950
DIVERS		
Dynamic Object C++	2066	2450
RTLink Plus	3693	4380
Tools.h ++ (avec source)	2266	2687
Brief C++	*nous consulter*	
CodeBase ++	1855	2200
Greenleaf Comm ++	1813	2150
Object Table C++ (avec source)	2943	3490
POET (Base de Données Objet sous Windows)	*nous consulter*	

BIBLIOTHEQUES / OUTILS C & PASCAL

POUR LANGAGE C	Frs HT	Frs TTC
C Tools +	1096	1300
GreenLeaf Superfunctions 1.1	2066	2450
Polylibrarian II	1981	2350
HALO Professional	3035	3600
Plink 86 +	2766	3280
Blinker	1813	2150

POUR PASCAL		
Turbo Vision Dev ToolKit	1433	1700
Object Professional 1.1	1518	1800
Turbo Professional 5.1	1180	1400
Turbo Magic 2.02	1813	2150
Turbo Analyst 6.0	1180	1400

SGBD

SGBD	Frs HT	Frs TTC
SQL Server V 1.11 10 postes	**11992**	**14223**
Oracle	*nous consulter*	
Paradox 3.5 Fr	4874	5780
Paradox Engine 2.0 US	3497	4147
Fox Pro Mono 2.0 Fr	5852	6940
Fox Base + 2.1 Fr	5299	6285

• 28, Bd de Strasbourg • 75010 PARIS • Tél. 40.18.10.70 • Fax 40.18.10.77 •

• Merci de bien vouloir préparer votre commande, format disquette et version logiciel, adresse de livraison et adresse de facturation, avant de nous appeler •
• Règlements par chèque à la commande ou par Contre-Remboursement (avec supplément) • Autres, nous consulter •

Frais de port France Métropolitaine : 80 F (95 F TTC) pour commande inférieure à 1200 F HT, 150 F (177,90 F TTC) pour commande de 1200 à 5000 F. Nous consulter pour commande supérieure

Service lecteurs n° 14 (dernière page)

EQUIP'CONTACT 40 22 09 23 R.C. PARIS B 338 013 154 (88B 02167)

TOUS NOS PRIX SONT EN FRANCS FRANCAIS ET SONT SUSCEPTIBLES DE MODIFICATIONS SANS PRÉAVIS • DANS LA LIMITE DES STOCKS DISPONIBLES •

Vous connaissez l'une de nos casquettes, découvrez les autres

EDITEUR DE LOGICIEL • FORMATION
INGENIERIE LOGICIELLE • RECHERCHE

Depuis sa création, NSL innove dans les domaines de l'interface utilisateur par le choix de technologies gagnantes: UNIX, le système de fenêtrage X et OSF/Motif. NSL a développé son savoir-faire autour de quatre axes: la conception de logiciels, la formation, l'ingénierie logicielle et la recherche.

Nos progiciels (XFM2, Wx2, WISh2, wi) sont leaders dans leur domaine avec entre autre le choix par Atari US Corp. d'en équiper ses stations UNIX. Numéro 1 en France dans la formation au système de fenêtrage X, à Motif et C++, NSL est aussi centre de formation PostScript agréé ADOBE. Notre expérience en ingénierie logicielle nous a valu d'être retenus pour des projets de très haute technicité, tel que le développement de l'interface graphique d'un logiciel de PAO.

L'ensemble de ces compétences permet aujourd'hui à NSL de proposer des produits et prestations de qualité. NSL est membre de l'OSF.

NSL
NON STANDARD LOGICS

57/59, RUE LHOMOND 75005 PARIS TEL: (1) 43 36 77 50 • FAX: (1) 43 36 59 78

UNIX, X ET OSF/Motif sont des marques déposées.

Les DIB ne peuvent pas être sélectionnées comme contextes de périphérique. Les diverses opérations graphiques de la GDI qui prennent un hDC comme argument (Rectangle, Ellipse, TextOut, etc.) ne peuvent pas fonctionner dans une DIB. Il faut convertir la DIB en bitmap dépendante, accomplir les opérations, puis reconvertir la bitmap en DIB. Néanmoins, cela peut provoquer une perte des informations graphiques. Par exemple, une DIB au format 24 bits perdra une partie de ses informations si elle est convertie en bitmap dépendante au format 4 bits.

Les polices et l'affichage du texte

Comme la plupart des interfaces graphiques, Windows supporte plusieurs polices de caractères (encore appelées «fontes»). Windows lui-même propose des polices telles que Helvetica, Courier, Roman et Symbol. Les applications Windows peuvent fournir des polices supplémentaires, et il est possible d'en créer avec le SDK Windows ou un éditeur de polices compatible. Comme nous l'avons déjà vu, la fonction standard d'affichage du texte dans une application Windows est TextOut. Elle prend comme arguments le handle d'un contexte d'affichage, les coordonnées logiques du début de l'affichage, la chaîne à afficher et la longueur de la chaîne. Exemple d'appel à TextOut :

```
TextOut(hDC,10,300,"Hello",5);
```

Le contexte d'affichage est important car il spécifie les caractéristiques qui seront appliquées au texte. Celles qui comprennent la couleur d'avant-plan et la couleur de fond du texte, le mode d'écriture du fond (opaque ou transparent), l'espacement entre les caractères et, bien sûr, la police utilisée. Lorsqu'on a obtenu un contexte de périphérique à l'aide de GetDC, la police System est associée à ce contexte. Cette police de Windows 3.0 est une police à espacement proportionnel (variable) et se compose du jeu de caractères Ansi. On peut observer un exemple de cette police en appelant directement TextOut après un appel à GetDC. Par défaut, les caractères sont affichés en noir sur un fond blanc.

```
HDC hDC;
hDC = GetDC(hWnd);
TextOut(hDC,10,300,"Hello",5);
// police System, caractères
   noirs sur fond blanc
ReleaseDC(hWnd, hDC);
```

Un caractère d'une police matricielle est en fait une petite bitmap composée de 1 et de 0. Le caractère proprement dit est constitué par les pixels à 1, le fond par les pixels à 0. Supposons qu'on écrive un émulateur de terminal et qu'on souhaite afficher les caractères en vert sur fond noir. La fonction SetTextColor permet de modifier la couleur des pixels de l'avant-plan (les pixels à 1) et la fonction SetBkColor change la couleur du fond (pixels à 0). Le second argument de ces deux fonctions est une valeur de couleur RGB. Pour produire des caractères verts sur un fond noir, il faut les instructions suivantes :

```
HDC hDC;
hDC = GetDC(hWnd);
SetTextColor(hDC,RGB(0x00,0xFF,  0x00));
                // vert
SetBkColor(hDC, RGB(0x00,0x00,  0x00));
                // noir
TextOut(hDC,10,300,"Hello", 5);
                // police System, caractères
                verts sur fond noir
ReleaseDC(hWnd, hDC);
```

Bien entendu, la couleur du fond (arrière-plan) n'est importante que si le mode d'arrière-plan est OPAQUE, et non TRANSPARENT. Le mode par défaut d'un contexte d'affichage par défaut est OPAQUE. Si le mode d'arrière-plan est modifié en TRANSPARENT, les pixels à 0 de la bitmap du caractère ne seront pas affichés sur l'écran. Le contenu courant de l'écran sera préservé à ces emplacements. On peut définir le mode d'arrière-plan par un appel à SetBkMode :

```
SetBkMode(hDC, TRANSPARENT);
```

L'emploi de différentes tailles et de divers styles de caractères proportionnels peut améliorer la présentation des applications Windows, même si l'application elle-même ne l'exige pas absolument (comme pourrait l'exiger un traitement de texte). D'autre part, certains programmes — tels qu'un émulateur de terminal — ont généralement besoin d'employer systématiquement une police à espacement fixe, dans laquelle chaque caractère à la même largeur. Si la taille de la police doit être redéfinie en fonction de la taille de la fenêtre, pour que celle-ci affiche toujours le même nombre de lignes et de caractères par ligne, il faut employer des polices proposant plusieurs tailles.

Les polices logiques

Il existe une différence entre une police physique et une police logique. Une police logique est une sorte de «liste de souhaits» indiquant les caractéristiques à donner à la

FIGURE 3 : AFFICHAGE D'UNE DIB

```
VOID PASCAL DisplayDIBitmap(HWND hWnd,
       HDC hDC, HANDLE hBits,
       int nWidth, int nHeight)
{
  HANDLE      hbmi;
  LPBITMAPINFO       lpbmi;
  RECT rClient;
  LPSTR       lpBits;

  if ((lpBits=GlobalLock(hBits)) == NULL)
  {
  MessageBox(hWndMain, "Impossible de
   verrouiller lpBits",     "DisplayDIB",
   MB_OK);
  return;
  }

  if ((hbmi= AllocateDefaultBitmap(nWidth,
   nHeight)) != NULL)
       lpbmi = (LPBITMAPINFO)
GlobalLock(hbmi);
  else
  {
       GlobalUnlock(hBits);
       MessageBox(hWndMain, "Impossible
      d'allouer la  bitmap par défaut",
        "DisplayDIB", MB_OK);
       return;
  }

/* Affiche la bitmap sur l'écran */

GetClientRect(hWnd, &rClient);
StretchDIBits(hDC, 0, 0, Client, right+1,
rClient, bottom+1,
       0, 0,
       (WORD) lpbmi ->bmiHeader.biWidth,
       (WORD) lpbmi ->bmiHeader.biHeight,
       lpBits, (LPBITMAPINFO) lpbmi,
       DIB_RGB_COLORS, SRCCOPY);

/* Déverrouiller tous les handles mémoire
*/

GlobalUnlock(hBits);
GlobalUnlock(hbmi);
GlobalFree(hbmi);
}

HANDLE PASCAL AllocateDefaultBitmap(int
nWidth, int Height)
{
HANDLE hbmi;
LPBITMAPINFO  lpbmi;
int     i;

static BYTE rgb[16][3] =
{  /*          B       G       R   */
       { 0x00, 0x00, 0x00 },
       { 0x00, 0x00, 0x80 },
       { 0x00, 0x80, 0x00 },
       { 0x00, 0x80, 0x80 },
       { 0x80, 0x00, 0x00 },
       { 0x80, 0x00, 0x80 },
       { 0x80, 0x80, 0x00 },
       { 0x80, 0x80, 0x80 },
       { 0xC0, 0xC0, 0xC0 },
       { 0x00, 0x00, 0xFF },
       { 0x00, 0xFF, 0x00 },
       { 0x00, 0xFF, 0xFF },
       { 0xFF, 0x00, 0x00 },
       { 0xFF, 0x00, 0xFF },
       { 0xFF, 0xFF, 0x00 },
       { 0xFF, 0xFF, 0xFF }
};

hbmi = GlobalAlloc(GMEM_FIXED |
GMEM_ZEROINIT,
       (DWORD) sizeof(BITMAPINFOHEADER) +
       16 * sizeof(RGBQUAD));
if (hbmi == NULL )
  return NULL;
lpbmi = (LPBITMAPINFO) GlobalLock(hbmi);

lpbmi->bmiHeader.biSize     = (long)
sizeof(BITMAPINFOHEADER);
lpbmi->bmiHeader.biWidth    = nWidth;
lpbmi->bmiHeader.biHeight   = nHeight;
lpbmi->bmiHeader.biPlanes   = 1;
lpbmi->bmiHeader.biBitCount        = 4;
lpbmi->bmiHeader.biCompression     = BI_RGB;
lpbmi->bmiHeader.biSizeImage       =((long)
nWidth * nHeight) / 2L;
lpbmi->bmiHeader.biXPelsPerMeter  = 0;
lpbmi->bmiHeader.biYPelsPerMeter  = 0;
lpbmi->bmiHeader.biClrUsed = 0:
lpbmi->bmiHeader.biClrImportant   = 0;

/ * Initialisation des couleurs */

for (i = 0; i < 16; i++)
       _fmemcpy(&lpbmi->bmiColors[i],
  rgb[i], 3);

GlobalUnlock(hbmi);
return hbmi;
}
```

police. Si l'on veut afficher du texte en Univers 24 points, en gras et en italique, il ne s'agit pas d'une police standard de Windows (bien qu'elle puisse éventuellement être fournie par un autre fabricant). Autrement dit, il n'existe pas de police physique correspondant à cette spécification. Mais Windows ne va pas

se bloquer : il s'efforce de trouver, parmi les polices physiques installées, une police dont les caractéristiques se rapprochent de celle demandée. Pour que cette corrélation automatique puisse être effectuée, il faut établir une requête de police logique. Windows stocke la requête et renvoie un handle à votre application.

Quand vient le moment d'associer cette requête à un contexte de périphérique, Windows essaie de transformer la requête de police logique en requête d'une police physique existante, à l'aide d'un algorithme de correspondance. Il fera tout son possible pour satisfaire la requête, même en modifiant l'aspect des polices existantes. Le résultat peut convenir ou non à ce qui a été demandé.

On peut employer deux fonctions pour établir une requête de police logique. Avec la fonction CreateFont, on décrit les propriétés de la police logique à l'aide d'une longue liste d'arguments. La fonction CreateFontIndirect prend comme argument une structure LOGFONT. Celle-ci contient les mêmes informations que celles qui peuvent être passées comme paramètres à CreateFont. Ces deux fonctions traitent les données contenues dans une structure LOGFONT *(voir figure 4)*. Certains champs de cette structure ne sont pas couramment utilisés (les champs lfEscapement et lfOrientation, par exemple) et peuvent prendre la valeur NULL.

Imaginons que nous demandions une police Times Roman gras 10 points. La figure 5 montre les deux méthodes de requête. Chacune des fonctions renvoie un handle sur une police. Ces appels ne chargent pas une police physique; Windows s'en occupe quand l'application veut associer hFont au contexte de périphérique - autrement dit, lors d'un appel à SelectObject.

Voici la manière habituelle d'afficher un texte avec une police non-système.

```
HDC hDC;
HFONT hFont, hOldFont;
hDC = GetDC(hWnd);
hFont = CreateFontIndirect((  LOGFONT FAR * )
&lfTimesRomanBold_10x8);
hOldFont = SelectObject(hDC,  hFont);
       //chargement ici
TextOut(hDC,10,300,"Hello",5);
SelectObject(hDC, hOldFont);
DeleteObject(hFont); ReleaseDC(hWnd, hDC);
```

La fonction SelectObject renvoie un handle sur l'objet précédent; dans ce cas, un handle sur la police système (System). Le handle de la police Times Roman est associé au handle du contexte d'affichage hDC. Tout texte envoyé vers hDC est affiché avec la nouvelle police. Après l'écriture de la chaîne, le contexte d'affichage doit être rendu au cache de contexte de Windows. Il faut d'abord désassocier le handle de la police Times Roman et le contexte d'affichage. Pour cela, on rappelle SelectObject en lui passant cette fois le handle de l'ancienne police système. Des compteurs associés aux objets GDI indiquent à Windows si un objet est utilisé. Pour libérer les ressources allouées à un objet GDI, il faut resélectionner l'ancien objet GDI dans le contexte de périphérique. Windows diminue alors de 1 le compteur de l'objet, et celui-ci est donc plus apte à être libéré. Il faut supprimer ensuite la police logique par un appel à DeleteObject.

Ce n'est pas obligatoire. Si l'application doit changer fréquemment de polices, il est possible de conserver le handle sur la police logique. Pour plus de détails sur la transformation et la correspondance des polices, on peut consulter le manuel «*SDK Reference*», volume 1, section 2.13.7, qui contient d'intéressantes informations à propos de l'algorithme de correspondance des polices. Il assigne un certain nombre de points de pénalité à chaque police, en gros pour chacune des caractéristiques de la police physique qui ne correspond pas à une caractéristique de la police logique. La police gagnante est celle qui a le moins de points de pénalité!

Informations sur les polices

La GDI Windows possède des fonctions permettant à une application d'obtenir des informations sur une police. Presque toutes ces fonctions ont pour format GetTextXxx. La fonction GetTextFace renvoie le nom de la police courante.

```
char szFace[80];
GetTextFace(hDC, sizeof(szFace),
  (LPSTR) szFace);
```

On peut demander les couleurs courantes et le mode d'arrière-plan à l'aide des fonctions GetBkColor, GetTextColor et GetBkMode.

```
COLORREF rgbBack, rgbText;
int iBkMode;

rgbBack         = GetBkColor(hDC); rgbText
                = GetTextColor(hDC); iBkMode
                = GetBkMode(hDC);
```

Comme les applications emploient souvent des polices à espacement proportionnel, il n'est pas aisé de calculer la largeur et la hauteur en pixels d'une chaîne donnée. Heureusement, la fonction GetTextExtent s'en

FIGURE 4 :

La structure LOGFONT définit les attributs d'une police, un objet graphique employé pour écrire du texte sur une surface d'affichage.

```
typedef struct tagLOGFONT {
        short   int     lfHeight;
        short   int     lfWidth;
        short   int     lfEscapement;
        short   int     lfOrientation;
        short   int     lfWeight;
        BYTE            lfItalic;
        BYTE            lfUnderLine;
        BYTE            lfStrikeOut;
        BYTE            lfCharSet;
        BYTE            lfOutPrecision;
        BYTE            lfClipPrecision;
        BYTE            lfQuality;
        BYTE            lfPitchAndFamily;
        BYTE            lfFaceNme[LF_FACESIZE];
} LOGFONT;
```

La structure LOGFONT possède les champs suivants :

CHAMP	DESCRIPTION
lfHeight	Spécifie la hauteur moyenne de la police en unités définies par l'utilisateur. La hauteur d'une police peut être spécifiée de trois manières. Si le champ lfHeight est supérieur à 0, il est transformé en unités de périphérique et adapté à la hauteur de caractère des polices disponibles. Si lfHeight est égal à 0, une taille par défaut est employée. si lfHeight est inférieur à 0, il est transformé en unités de périphérique et la valeur absolue est adaptée à la hauteur de caractère des polices disponibles.
lfWidth	Spécifie la largeur moyenne des caractères de la police, en unités périphériques. Si le champ lfWidth est à 0, le facteur de correction des proportions du périphérique est adapté au facteur de correction des polices disponibles, en cherchant la plus petite différence en valeur absolue.
lfEscapement	Spécifie, en 1/10 de degré, l'angle entre le vecteur d'échappement et l'axe des x de la surface d'affichage. Le vecteur d'échappement est la ligne passant par les origines du premier et du dernier caractère d'une ligne de texte. L'angle est mesuré dans le sens inverse des aiguilles d'une montre, à partir de l'axe des x.
lfOrientation	Spécifie, en 1/10 de degré, l'angle entre la ligne de base d'un caractère et l'axe des x. L'angle est mesuré dans le sens inverse des aiguilles d'une montre, à partir de l'axe des x.
lfWeight	Spécifie la «graisse» (l'épaisseur) de la police, en nombre de pixels «encrés» pour 1 000. Bien que le champ lfWeight puisse être un entier compris entre 0 et 1 000, les valeurs courantes sont les suivantes : 400 Normal 700 Gras Ces valeurs sont approximatives; l'aspect réel dépend de la police. Si lfWeight a pour valeur 0, une épaisseur par défaut est utilisée.
lfItalic	Spécifie une police en italique, si la valeur est différente de 0.
lfUnderline	Spécifie une police soulignée, si la valeur est différente de 0.
lfStrikeOut	Spécifie une police en barré, si la valeur est différente de 0.
lfCharSet	Spécifie le jeu de caractères de la police. Les trois valeurs prédéfinies sont : ANSI_CHARSET OEM_CHARSET SYMBOL_CHARSET Le jeu de caractères OEM est dépendant du système, qui peut proposer des polices ayant d'autres jeux de caractères. Si une application emploie une police ayant un jeu de caractères inconnu, elle ne devrait pas essayer de traduire ni d'interpréter les chaînes affichées avec cette police, mais les passer directement au pilote du périphérique de sortie.
lfOutPrecision	Spécifie la précision de la police en sortie, qui définit jusqu'à quel point la sortie doit correspondre aux valeurs demandées pour la hauteur, la largeur, l'orientation des

charge. Elle renvoie les dimensions d'une chaîne donnée affichée avec la police courante. La hauteur est renvoyée dans le mot de poids fort et la largeur dans le mot de poids faible.

```
DWORD dwDim;
WORD yHeight, xWidth;
dwDim = GetTextExtent(hDC,(LPSTR)  "Hello", 5);
yHeight = HIWORD(dwDim);
xWidth = LOWORD(dwDim);
```

La fonction la plus importante est peut-être GetTextMetrics qui est employée par presque toutes les applications Windows, même si elles n'utilisent que la police System. GetTextMetrics remplit une structure TEXTMETRIC avec les informations relatives à la police courante. Ces informations sont différentes de celles que contient la structure LOGFONT, car elles se rappor-

LA STRUCTURE LOGFONT

caractères, l'échappement et l'espacement. La valeur par défaut est OUT_DEFAULT_PRECIS.

lfClipPrecision — Spécifie la précision de «clipping» de la police, qui définit comment couper des caractères situés partiellement en dehors de la région de clipping (zone d'affichage). La valeur par défaut est : CLIP_DEFAULT_PRECIS.

lfQuality — Spécifie la qualité de la police en sortie, qui définit le soin apporté par la GDI pour faire correspondre les attributs d'une police logique à ceux d'une police physique. Les valeurs possibles sont :

Valeur	Signification
DEFAULT_QUALITY	L'aspect de la police n'est pas important.
DRAFT_QUALITY	L'aspect de la police est moins important que si la valeur PROOF_QUALITY est spécifiée. Pour les polices GDI, la transformation est autorisée, ce qui signifie que davantage de polices sont disponibles, mais au détriment de la qualité. Des polices grasses, italiques ou barrées sont créées en cas de besoin.
PROOF_QUALITY	La qualité des caractères est plus importante que la correspondance exacte avec les attributs de la police logique. Pour les polices GDI, la transformation est interdite et la police ayant la taille la plus proche est sélectionnée. Bien que la taille de la police choisie puisse être différente lorsque PROOF_QUALITY est spécifiée, la qualité du «rendu» de la police est élevée et elle ne présente pas de distorsion. Des polices grasses, italiques ou barrées sont créées en cas de besoin.

lfPitchAndFamily — Spécifie l'espacement et la famille de la police. Les deux bits de poids faible indiquent l'espacement et peuvent prendre l'une des valeurs suivantes :
DEFAULT_PITCH
FIXED_PITCH
VARIABLE_PITCH
Les quatre bits de poids fort du champ indiquent la famille de la police et peuvent prendre l'une des valeurs suivantes :
FF_DECORATIVE
FF_DONTCARE
FF_MODERN
FF_ROMAN
FF_SCRIPT
FF_SWISS
La valeur désirée peut être obtenue en utilisant l'opérateur booléen OU pour associer une constante d'espacement à une constante de famille. Les familles de polices décrivent l'aspect général d'une police. Elles servent à spécifier un type de police lorsque la police souhaitée n'est pas disponible.
Les significations de ces constantes sont les suivantes :

Valeur	Signification
FF_DECORATIVE	Police fantaisie.
FF_DONTCARE	Sans importance ou inconnue.
FF_MODERN	Polices à espacement fixe, avec ou sans empattement. Ces polices sont généralement modernes.
FF_ROMAN	Polices à espacement proportionnel, avec empattement.
FF_SCRIPT	Polices dont l'aspect ressemble à l'écriture manuelle.
FF_SWISS	Polices à espacement proportionnel, sans empattement.

lfFaceName — Spécifie le type de la police. Il doit s'agir d'une chaîne de caractères à zéro terminal. Si lfFaceName a pour valeur NULL, la GDI utilise un type par défaut.

tent à la police physique courante et non à la police logique *(voir figure 6)*. Les membres les plus importants de la structure TEXTMETRIC sont les champs indiquant la largeur et la hauteur des caractères de la police.

La structure TEXTMETRIC est essentielle en raison de la nature graphique de Windows. Presque tous les programmes écrivent sur l'écran, et l'affichage contient généralement plusieurs lignes d'informations. Comment le curseur est-il avancé au début de la prochaine ligne physique de l'écran afin de préparer la position de l'affichage du texte?

Normalement, une application doit envoyer des caractères «retour-chariot/nouvelle ligne» à la fin d'une ligne. Si l'application emploie un système de fenêtre en mode texte, il lui suffit de spécifier les coordonnées des caractères de l'écran, puisque la largeur et la hauteur de chaque caractère est de 1 unité. Dans le cas d'une inter-

FIGURE 5 :
2 FAÇONS DE DEMANDER UNE POLICE

```
HFONT hFont;

hFont = CreateFont( 10,     //hHeight
 8,                         //nWidth
NULL,                       //nEscapement
NULL,                       //nOrientation
700,                        //nWeight,700 == bold
FALSE,                      // cItalic
FALSE,                      // cUnderlined
FALSE,                      // cStrikeout
ANSI_CHARSET,               // cCharSet
OUT_DEFAULT_PRECIS,
       //cOutPutPrecision
CLIP_DEFAULT_PRECIS,
       //cClipPrecision
DEFAULT_QUALITY,
       //cQuality
VARIALE_PITCH | FF_ROMAN,
       //cPitchAndFamily
(LPSTR) "Tms Rmn"
       //lpFacename
       );


LOGFONT lfTimesRomanBold_10x8 =
{
10,             // hHeight
8,              // nWidth
NULL,           //nEscapement
NULL,           // nOrientation
700,            // nWeight, 700 == bold
FALSE,          // cItalic
FALSE,          // cUnderlined
FALSE,          // cStrikeout
ANSI_CHARSET,           // cCharSet
OUT_DEFAULT_PRECIS,
                // cOutPutPrecision
CLIP_DEFAULT_PRECIS,
                // cClipPrecision
DEFAULT_QUALITY,
                // cQuality
VARIALE_PITCH | FF_ROMAN,
                // cPitchAndFamily
(LPSTR) "Tms Rmn"
                // lpFacename
};

hFont = CreateFontIndirect(
       (LOGFONT FAR *)
&lfTimesRomanBold_10x8);
```

face graphique, l'écran est adressé en pixels, et les choses sont bien différentes. Les caractères d'une même police peuvent avoir une largeur ou une hauteur spécifique, différentes de celles des autres caractères.

En outre, un caractère peut être positionné sur n'importe quel pixel. Pour préparer l'affichage d'une nouvelle ligne, il ne suffit pas de demander au système d'avancer d'une unité.

Il faut déterminer la hauteur en pixels de la police courante, et avancer verticalement de cette hauteur (la situation se complique encore si vous utilisez sur une même ligne plusieurs polices ayant des tailles différentes, mais nous ne considérerons pas un tel cas pour le moment).

La hauteur d'un caractère est l'une des informations contenues dans la structure TEXTMETRIC. Pour placer dans une structure de ce type des indications relatives à une police, nous employons la fonction GetTextMetrics :

```
TEXTMETRIC tm;
GetTextMetrics(hDC,(LPTEXTMETRIC)
  &tm);
```

Une fois l'information récupérée, il est facile de déterminer où doit débuter la prochaine ligne. Le champ tmHeight contient la hauteur de chaque caractère. La valeur tmExternalLeading représente l'interligne que propose le concepteur de la police.

Pour calculer la position du pixel sur lequel doit se placer le prochain caractère, on peut employer cette simple équation :

```
yNextLine = yCurrentLine + tm.tmHeight +
tm.tmExternalLeading
```

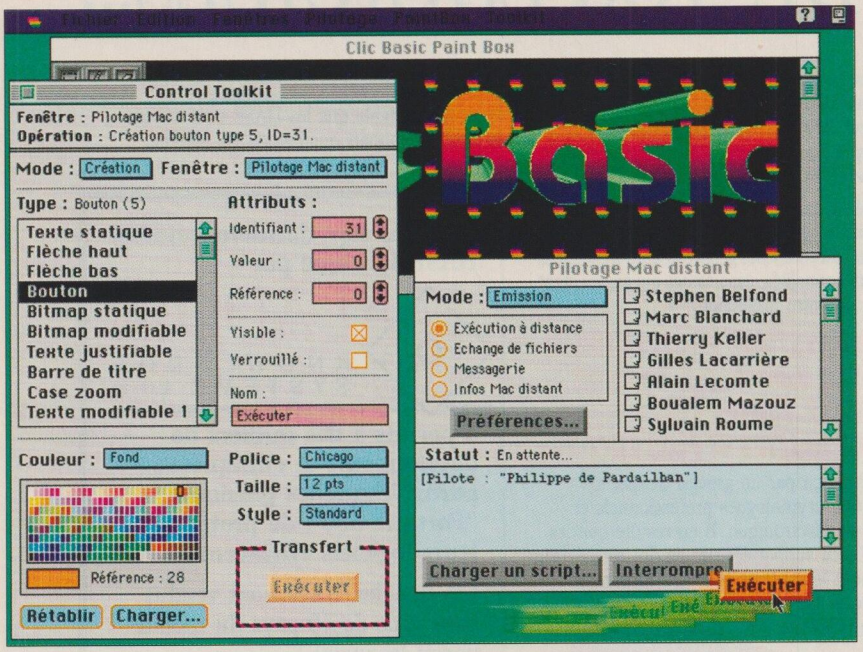
Voici un exemple de code permettant d'écrire plusieurs lignes de texte :

```
int yPos;
int yIncrement;
int iLine;
TEXTMETRIC tm;
GetTextMetrics(hDC,(LPTEXTMETRIC)
  &tm);
yIncrement = tm.tmHeight +
tm.tmExternalLeading;

  for   (ypos = iLine = 0;
        iLine < nMaxLines;
         yPos +=yIncrement,iLine++)
         {
         TextOut(hDC,0,yPos,szText[iLine],
         strlen(szText[iLine]));
         }
```

Technotruffe présente un logiciel signé Philippe de Pardailhan.

La création d'applications Macintosh pour tous.



CLIC BASIC™

Clic Basic, pour qui ?

Vous êtes un pro de la programmation...

- Avec Clic Basic, proposez des applications ouvertes que les utilisateurs et les SSII pourront faire évoluer.
- Récupérez toutes vos XCMD et XFCN d'HyperCard™.

Vous rêvez de porter vos applications PC, ou simplement de programmer, sous MAC/OS...

- Consacrez le meilleur de votre temps à peaufiner vos programmes, Clic Basic se charge de l'interface.
- Importez vos externales C, Pascal, etc.

Vous avez des rudiments de Basic...

- Impressionnez-vous et créez une application Mac en quelques coups de souris, en quelques lignes d'un langage aussi puissant que simple.
- La "Toolbox", "QuickDraw", les "Apple Events", toutes les ressources du Mac pour la première fois à votre disposition !

Vous ne voulez surtout pas manquer la dernière révolution de la Mac-Culture...

- Découvrez des dizaines d'applications, de jeux, d'utilitaires Clic Basic et comprenez dès aujourd'hui ce que sera la micro-informatique de demain.

Clic Basic, comment ?

L'interface Mac... à la souris

- Bibliothèque de dizaines de fenêtres et de contrôles pré-programmés, tous redéfinissables.
- Gestion des polices, des couleurs, des sons, etc.
- Importation d'icônes, de PICTs.
- Création et exploitation de bases de données.

Le Mac maîtrisé

- Multitâche, Clic Basic gère simultanément autant de programmes autonomes que permet votre mémoire.
- Clic Basic permet d'intercepter tous les événements utilisateur : souris, fenêtres, clavier, menus ; tous les événements système...
- Clic Basic gère les communications inter-machines, les périphériques, les réseaux (Apple Events, IAC, PrintManager, etc.).

Un langage simple, complet et rapide

- Plus de 400 instructions pour piloter l'affichage, les contrôles, les dialogues, les événements, les fichiers, les fenêtres, les couleurs, les ressources, les pictures, les icônes, l'impression, les entrées/sorties, les listes, les menus, les processes, les sons, les textes, etc.
- Génération d'applications semi-compilées.
- SGBD C-Tree+™ intégré.

Un environnement de programmation convivial

- Aide en ligne : base de données de toutes les instructions commentées (avec insertion automatique).
- Editeur de scripts évolué : indentation automatique, mise en forme, recherche/remplacement automatique ou manuel, analyse et déboguage.
- Tests et exécution en temps réel.

Devenez partenaire privilégié Clic Basic.

Offre exceptionnelle de lancement*

899 F H.T. au lieu de 1500 F H.T.

Hot-line directe avec Philippe de Pardailhan

Kit de communication (avec câble) pour piloter le serveur Clic Basic : forums, téléchargement, jeux de réseaux, messagerie, etc.

Inscription au Clic d'Or '92 (10 000 F de dotation)

1 an de mises à jour gratuites

et bien sûr :

Plus de 30 programmes opérationnels fournis

Bibliothèque d'icônes, de fenêtres, de boutons

Manuels d'utilisation complets

*Offre limitée aux 500 premières réponses.

OUI, je souhaite devenir partenaire Clic Basic.

Nom : _____ Société : _____

Adresse : _____

Ville : _____ Code postal : _____

Nombre d'exemplaires Clic Basic : ____ x 1066,21 F TTC, soit : _____ F

Ci-joint, mon règlement à l'ordre de Technotruffe par :

❒ Chèque bancaire ❒ Chèque postal (3 volets) ❒ Carte Visa

Visa n° |_|_|_|_| |_|_|_|_| |_|_|_|_| |_|_|_|_| Expire fin : |_|_|_|_|

Date et signature

Bon de commande à retourner à
TECHNOTRUFFE
12, rue du Croissant, 75002 Paris

M.B. PHOSPHENES

AVANT DE CHOISIR UNE PROTECTION DE VOS LOGICIELS...

...contre le piratage et l'utilisation frauduleuse, vérifiez que le système de protection envisagé possède bien les qualités suivantes :

UNE CLÉ AU HARDWARE EXCEPTIONNEL

Les protections hardware sont maintenant devenues un standard incontournable dans le monde entier. Cependant toutes les clés ne sont pas égales. Une clé se doit de posséder les qualités suivantes :

✔ Compatibilité et transparence. La clé doit pouvoir fonctionner sans problème sur tous les ordinateurs de vos clients. L'utilisateur devrait pouvoir l'oublier dès sa connection.

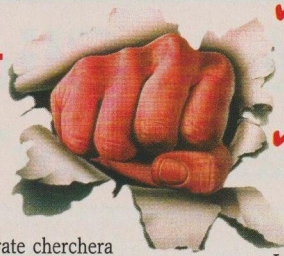
✔ Une électronique sans faille. Un circuit ASIC (spécifiquement conçu et fabriqué) devrait être inclu dans le hardware. Seul un tel circuit peut rendre impossible le "reverse engineering" et le piratage de la clé.

✔ Un code développeur unique et inaccessible doit être gravé dans l'ASIC. Le code client ne devrait jamais être stocké dans la mémoire de la clé où il peut être lu et modifié à volonté... Seul le code de l'ASIC permet une reconnaissance des messages.

✔ Une zone de mémoire disponible en lecture/écriture doit être disponible sur demande. On doit pouvoir écrire dans cette mémoire depuis n'importe quel PC, sans avoir besoin d'équipement spécial (sérialisation chez le client).

✔ Une consommation de courant très faible permet à la clé de fonctionner dans les pires conditions, du PC au laptop, avec ou sans imprimante connectée, allumée ou non...

UN LOGIGIEL PERFORMANT



Comme il est pratiquement impossible de dupliquer ou de pirater une clé qui possèderait toutes les qualités citées, le pirate cherchera la faille au niveau du logiciel protégé. Et afin de prévenir cela, le logiciel de protection se doit de posséder aussi les qualités suivantes :

✔ Un Module de Protection à Linker (MPL) intégré à votre programme source, afin de pouvoir insérer des appels à la clé depuis n'importe quel endroit du programme protégé.

✔ Un système "d'enveloppe" autour de votre programme exécutable. Votre programme protégé peut être verrouillé par cryptage et ne sera décrypté puis exécuté que si la présence de la clé est vérifiée...

✔ Des mécanismes sophistiqués de protection, des pièges antidébugging, des cryptages/décryptages dynamiques, du code auto-généré et auto-modifiable sont des fonctionnalités importantes qui augmentent la sécurité de votre protection.

HASP, LE SYSTÈME PROFESSIONNEL D'ALADDIN Knowledge Systems VOUS OFFRE TOUTES CES FONCTIONNALITÉS ET PLUS ENCORE :

Hasp a été conçu par un groupe d'experts informatiques, cryptologues professionnels et ingénieurs en électronique. Il en résulte que les clés HASP sont supportées par ce qui est probablement le meilleur software disponible sur le marché. De plus, le système HASP a été testé avec succès sur une vaste gamme d'ordinateurs. De surcroît, HASP vous offre :

✔ Le système FAS (Full Authorization System) qui permet de protéger avec une seule clé MemoHASP-4 près de 112 progiciels.

✔ Le système PCS (Pattern Code Security) permettant de mettre à jour "magiquement" les différentes valeurs de contrôle et qui évite les faiblesses d'une protection séquentielle.

✔ Un système de détection de virus qui peut être incorporé dans le programme protégé.

✔ Les HASP sont compatibles avec un grand nombre de systèmes d'exploitation comme DOS, SCO Xenix, et Unix-386, OS/2, AIX, PHAR-LAP, ERGO et RATIONAL DOS Extenders et aussi avec WINDOWS et AUTOCAD.

✔ NetHASP, la protection ultime : protégez les licences de dizaines de logiciels DOS ou Windows 3.0 fonctionnant en réseau avec une seule clé. Les réseaux supportés sont NOVELL (dédié ou non), Lan Manager, Lantastic, Banyan, DLink et tous les logiciels basés sur Net BIOS

LORS D'UN TEST EXÉCUTÉ DANS UN LABORATOIRE INDÉPENDANT ALLEMAND, LE HASP ETAIT LA SEULE CLÉ A NE PAS ETRE DÉPLOMBÉE.

(CT Magazine, RFA, Mai 1990)

ET FINALEMENT :

Il est indiscutable que les HASP vous offrent le meilleur rapport qualité/prix du marché.

Dans les sept dernières années, HASP fabriqué par

ALADDIN KNOWLEDGE SYSTEMS LTD.

a permis à des milliers de développeurs et entreprises, parmi lesquelles plusieurs "Fortune 500", de protéger leurs logiciels efficacement.

Les dongles HASP sont vendus dans plus de 40 pays du monde parmi lesquels :

- **Australie** Conlab Pty. Ltd., Tél. 3 8985685 - Fax: 3 8995759
- **Belgique** Akkermans bvba, Tél. 3 2338826, - Fax: 3 2315438
- **Tchécoslovaquie** ATLAS Ltd, Tél. + Fax: 2 821280
- **Danemark** SC Metric a/s, Tél. 42 804200 - Fax: 42 804131
- **Allemagne** CSS GmbH, Tél. 201 749860, - Fax: 201 748644
- **Grèce** Unibrain SA, Tél. 1 6856320, Fax: 1 6474943
- **Hollande** Akkermans BV, Tél. 45 241444 - Fax: 45 245515
- **Italie** Partner Data S.r.l., Tél. 2 33101709 - Fax: 2 347564
- **Corée** Hanil System Inc., Tél. 2 5639161 - Fax: 2 5538079
- **Nouvelle Zélande** Training Solutions Tél. 4 666014 - Fax: 4 697190
- **Pologne** Systherm Tél. 061 45065 - Fax: 061 324134
- **Portugal** Futurmatica Lda., Tél. 1 4116269 - Fax: 1 4116277
- **Espagne** PC Hardware Tél. 3 2493193 - Fax: 3 3337497
- **Suisse** Opag AG, Tél. 61 7112245 - Fax : 61 7115355

DISTRIBUTEUR POUR LA FRANCE

LOGIDATA SA

B.P. n° 2 - Publier - 74500 EVIAN / France
Tél. 50 70 73 75 - Fax 50 75 31 44

FIGURE 6 : LA STRUCTURE TEXTMETRIC

La structure TEXTMETRIC contient des informations sur une police physique. Toutes les tailles sont données en unités logiques (dépendant du mapping mode courant du contexte).

```
typedef struct tagTEXTMETRIC {
        short   int     tmHeight;
        short   int     tmAscent;
        short   int     tmDescent;
        short   int     tmInternalLeading;
        short   int     tmExternalLeading;
        short   int     tmAveCharWidth;
        short   int     tmMaxCharWidth;
        short   int     tmWeight;
        BYTE            tmItalic;
        BYTE            tmUnderlined;
        BYTE            tmStruckOut;
        BYTE            tmFirstChar;
        BYTE            tmLastChar;
        BYTE            tmDefaultChar;
        BYTE            tmBreakChar;
        BYTE            tmPitchAndFamily;
        BYTE            tmCharSet;
        short   int     tmOverhang;
        short   int     tmDigitizedAspectX;
        short   int     tmDigitizedAspectY;
} TEXTMETRIC;
```

La structure TEXTMETRIC possède les champs suivants :

tmHeight
Spécifie la hauteur totale des caractères (tmAscent + tmDescent).

tmAscent
Spécifie l'ascendant (hauteur des lettres montantes) : nombre d'unités au-dessus de la ligne de base.

tmDescent
Spécifie le descendant : nombre d'unités au-dessous de la ligne de base pour les lettres descendantes.

tmInternalLeading
Spécifie la hauteur de la tête des caractères (dans les limites indiquées par tmHeight). Les accents et certains caractères étrangers se placent dans cette zone. Le concepteur de la police peut donner la valeur 0 à ce champ.

tmExternalLeading
Spécifie l'espacement que l'application place entre les lignes de texte. Comme cette zone est extérieure à la police, elle ne contient aucun signe et ne sera pas altérée par l'affichage des caractères, ni en mode OPAQUE, ni en mode TRANSPARENT. Le concepteur de la police peut donner la valeur 0 à ce champ.

tmAveCharWidt
Spécifie la largeur moyenne des caractères de la police (définie grossièrement comme étant la largeur de la lettre x). Cette valeur n'inclut pas la largeur supplémentaire indispensable pour les caractères gras ou italiques.

tmMaxCharWidth
Spécifie la largeur du plus large caractère de la police.

tmWeight
Spécifie la graisse (l'épaisseur) de la police.

tmItalic
Spécifie une police en italique, si la valeur est différente de 0.

tmUnderlined
Spécifie une police soulignée, si la valeur est différente de 0.

tmStruckOut
Spécifie une police en barré, si la valeur est différente de 0.

tmFirstChar
Spécifie la valeur du premier caractère défini dans la police.

tmLastChar
Spécifie la valeur du dernier caractère défini dans la police.

tmDefaultChar
Spécifie la valeur du caractère qui sera substitué aux caractères n'existant pas dans la police (caractère de remplacement).

tmBreakChar
Spécifie la valeur du caractère utilisé pour signaler les coupures de mots (césures) dans un texte justifié.

tmPitchAndFamily
Spécifie l'espacement et la famille de la police. Le bit de poids faible représente l'espacement. S'il est à 1, l'espacement est variable. S'il est à 0, l'espacement est fixe. Les quatre bits de poids fort représentent la famille de la police. Le champ tmPitchAndFamily peut être combiné avec la valeur hexadécimale F0H à l'aide de l'opérateur ET avant d'être comparé aux noms des familles de polices.

tmCharSet
Spécifie le jeu de caractères de la police.

tmOverhang
Spécifie la largeur supplémentaire qui doit être ajoutée à une chaîne pour des caractères synthétiques. Lorsqu'elle crée certains attributs, tels que le gras ou l'italique, la GDI (ou le périphérique) doit augmenter la largeur de la chaîne en tenant compte à la fois des caractères et de la chaîne. A titre d'exemple, pour mettre une chaîne en gras, la GDI augmente l'espace entre les caractères et répète la chaîne avec une valeur de décalage, pour l'italique, les caractères de la chaîne sont inclinés. Dans les deux cas, la largeur de la chaîne est augmentée. Pour les chaînes en gras, l'augmentation par caractère correspond à la valeur de décalage. Pour les chaînes en italique, la largeur supplémentaire par caractère est la valeur de décalage du sommet par rapport à la base du caractère. Le champ tmOverhang permet à une application de déterminer quelle proportion de la largeur d'un caractère — retournée par GetTextExtent appliquée à un seul caractère — constituera la largeur réelle du caractère. De la même manière, l'application pourra déterminer la largeur supplémentaire d'une chaîne. La largeur réelle correspond à l'extension horizontale moins la largeur ajoutée.

tmDigitizedAspectX
Spécifie la résolution horizontale du périphérique pour lequel la police a été conçue.

tmDigitizedAspectY
Spécifie la résolution verticale du périphérique pour lequel la police a été conçue. Le rapport des champs tmDigitizedAspectX et tmDigitizedAspectY est le facteur de correction des proportions de ce périphérique.

La variable iLine donne la position des lignes de texte contenues dans un tampon. La variable yPos est employée pour conserver la position verticale du pixel où doit commencer la ligne courante. Après l'affichage d'une ligne de texte, yPos est incrémentée pour donner la prochaine position verticale. Cette boucle peut être optimisée en comparant yPos à la coordonnée y du bas de la fenêtre, afin de ne pas appeler inutilement TextOut puisque les lignes situées sous le bord inférieur de la fenêtre ne seront pas affichées.

Liste des polices

Une amélioration agréable de Windows est la possibilité pour chaque application de choisir parmi toute une variété de styles et de tailles. Dans notre application d'exemple (portefeuille d'actions), l'utilisateur doit pouvoir choisir le style et la taille des caractères utilisés pour la légende d'un graphe d'action.

Dans ce but, il faut demander à Windows une liste de toutes les polices supportées par le périphérique de sortie. Il existe heureusement une fonction toute prête : EnumFonts. Comme avec la plupart des fonctions d'énumération de Windows, il faut fournir à EnumFonts l'adresse d'une fonction «callback» qui sera appelée pour chaque police. L'information passée à la fonction callback, pour chacune des polices supportées, contient une structure LOGFONT et une structure TEXTMETRIC qui peuvent être employées pour la création et l'utilisation de la police logique. Le format d'EnumFonts est le suivant :

```
EnumFonts(hDC, lpFacename,  lpFontFunc, lpData);
```

Le second argument est une chaîne spécifiant le nom de la police à énumérer. Si l'argument est NULL, Windows sélectionne une seule police pour chaque «œil de caractère» (aspect, type de caractère) supporté par le périphérique et énumère cette police. Le troisième argument, lpFontFunc, est une FARPROC sur la fonction d'énumération. Comme toutes les fonctions callback, elle doit être exportée dans le fichier DEF de l'application, et il faut employer la fonction MakeProcInstance afin de créer un «thunk» de la fonction callback *(voir Langages & systèmes n°5).*

FIGURE 7 : ÉNUMÉRATION DES POLICES

```
#define ENUMFACES  0
#define ENUMSIZES  1

static int   idxTypeface;
static int   idxSize;
static BYTE  pbSizeMask;

VOID PASCAL GetTypefaces(HWND hWnd)
{
  FARPROC lpEnumFunc;
  HDC hDC;
  idxTypeface = 0;
  hDC = GetDC(hWnd);
  lpEnumFunc = MakeProcInstance(EnumFunc,
   hInstance);
  EnumFonts(hDC, (LPSTR) NULL,
       lpEnumFunc, (LPSTR) ENUMFACES);
  FreeProcInstance(lpEnumFunc);
  ReleaseDC(hWnd, hDC);
}

VOID PASCAL GetFontSizes(HWND hWnd, int
idxFont)
{
  HDC hDC;
  FARPROC lpEnumFunc;
  idxSize = 0;
  pbSizeMask =
FontInfo[idxFont].fSizeMask;
  hDC = GetDC(hWnd);
  lpEnumFunc = MakeProcInstance(EnumFunc,
   hInstance);
  EnumFonts(hDC,
   FontInfo[idxFont].szTypeface,
   lpEnumFunc, (LPSTR) ENUMSIZES);
  FreeProcInstance(lpEnumFunc);
  ReleaseDC(hWnd, hDC);
}
int FAR PASCAL EnumFunc(LPLOGFONT
 lpLogFont,  LPTEXTMETRIC lpTextMetric,
 short FontType, LPSTR lpData)
{
  switch ((int) lpData)
  {
    case ENUMFACES :
      if (idxTypeface >= MAXFONT)
        return 0;
_lstrcpy(FontInfo[idxTypeface].szTypeface,
            lpLogFont->lfFaceName);
      memset((char *)
&FontInfo[idxTypeface].fSizeMask, 0,
            sizeof(SIZEMASK));
      return ++idxTypeface;
     case ENUMSIZES :
      if (idxSize >= MAXSIZE)
       return 0;
      SetBit(pbSizeMask, lpLogFont-
>lfHeight);
      return ++idxSize;
  }
}
```

Le dernier argument est une valeur sur 32 bits qui peut représenter une donnée définie par l'application (par exemple, un handle sur un bloc mémoire où sont stockées toutes les informations concernant la police). Le format de la fonction d'énumération définie par l'application est le suivant :

```
int FAR PASCAL FontFunc(lpLogFont,
        lpTextMetrics,nFontType,lpData)
```

Les deux premiers arguments sont des pointeurs longs sur les structures LOGFONT et TEXTMETRIC. Le troisième, nFontType, est un masque de bits indiquant s'il s'agit d'une police vectorielle ou matricielle, et si elle dépend du périphérique ou de la GDI. Enfin, lpData est la valeur qui a été passée comme dernier argument à la fonction EnumFonts.

Le code de la figure 7 peut être employé pour énumérer tous les types et toutes les tailles disponibles. L'astuce consiste à utiliser une seule fonction d'énumération qui est invoquée par deux fonctions différentes — une pour le type de caractère, l'autre pour la taille.

La valeur du paramètre lpData indique à la fonction d'énumération qu'elle doit traiter soit le nom, soit la taille. Le nom de chaque police est stocké dans une mythique structure intitulée FontInfo, avec un masque de bits représentant les points disponibles pour la police en question. Un bit à 1 dans le masque signifie que la taille correspondante est disponible. Par exemple, si une police 10 points est libre, le bit 10 du masque est mis à 1. Comme une police peut proposer jusqu'à 127 points, nous pouvons représenter toutes les tailles possibles à l'aide d'un tableau de 16 octets.

Après avoir compilé une liste de types et de tailles, il est facile de placer ces informations dans une boîte liste ou dans un menu déroulant multiniveaux pour que l'utilisateur puisse sélectionner une police à tout moment. Il faut mentionner ici le cas des polices vectorielles. Windows peut déterminer n'importe quelle taille pour une police vectorielle. On peut donc créer une police de taille quelconque, même si elle n'est pas indiquée dans le tableau.

Polices personnalisées

Le Font Editor du Windows SDK permet de créer des polices personnalisées lorsqu'on souhaite employer un nouveau type de caractères dans une application *(voir figure 8)*. L'une des raisons de créer ou de modifier une police est tout simplement que certains caractères ne sont pas disponibles dans les polices existantes. Considérons notre portefeuille d'actions. Pour l'instant, la seule façon de représenter les fractions consiste à placer un point devant la partie décimale. La fraction 5/8 sera présentée sous la forme .625. Il serait plus esthétique de représenter cette fraction par un seul caractère. La bitmap comporterait un petit 5 dans la partie supérieure, une barre oblique, et un petit 8 dans la partie inférieure .

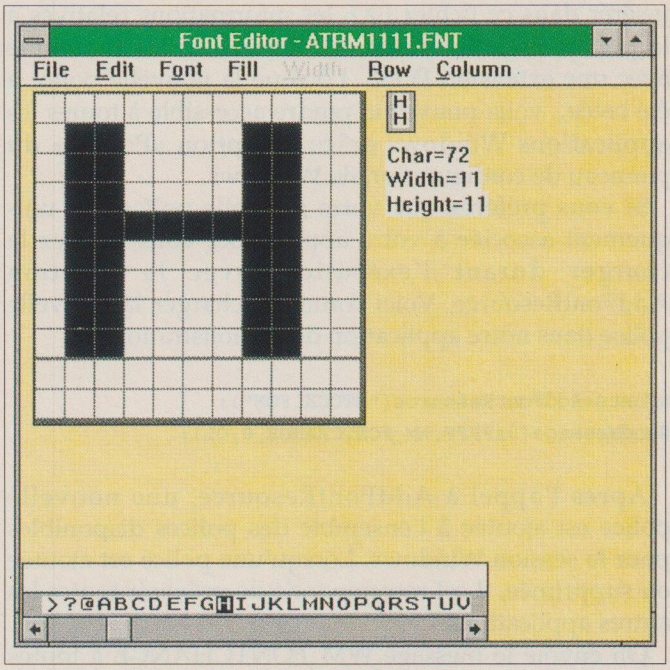


Figure 8 : création d'une police avec le SDK Font Editor.

Le Font Editor est fourni avec deux fichiers FNT qui peuvent être employés comme base pour créer une nouvelle police. ATRM1111.FNT constitue une police système à espacement fixe, VGASYS.FNT est une police à espacement proportionnel. Pour fabriquer les caractères de fraction, on peut modifier l'un de ces fichiers et remplacer certains caractères inutilisés. Dans le jeu de caractères Ansi, les vingt premiers caractères (environ) au-dessus du 127 ne sont généralement pas utilisés. Vous pouvez prendre comme modèles les trois caractères de fraction Ansi déjà existants : , et _.

Bien entendu, la police modifiée devra être sauvegardée sous un autre nom. Il y a une différence entre un fichier FNT et un fichier FON. Une application Windows peut employer un fichier FON, mais pas un fichier FNT. La section 18.12 du *«SDK Guide to Programming»* indique la démarche à suivre pour transformer en fichier FON la description de la police contenue dans un fichier FNT. Il faut d'abord créer un très petit fichier en langage d'assemblage et le compiler en fichier objet, sans code ni données.

Il faut encore établir un fichier RC contenant une liste des fichiers FNT qui contiennent la police. L'éditeur de liens produit un fichier EXE à partir du petit fichier objet. Le compilateur de ressources est alors utilisé pour

insérer dans ce fichier EXE les informations relatives à la police. Pour terminer, le fichier EXE est renommé avec une extension FON. Lorsque la nouvelle police a été créée, vous pouvez la rendre accessible à toutes les applications Windows grâce à l'option «Polices» du panneau de configuration de Windows.

Si vous préférez que votre nouvelle police soit uniquement associée à votre application, vous pouvez la charger durant l'exécution, avec la fonction AddFontResource. Voici comment charger la nouvelle police dans notre application de démonstration :

```
nFonts=AddFontResource("STOCK.FON");
SendMessage(0xFFFF,WM_FONTCHANGE,0,0L);
```

Après l'appel à AddFontResource, une nouvelle police est ajoutée à l'ensemble des polices disponibles pour la session Windows. Lorsqu'une police est ajoutée ou supprimée, il est recommandé de prévenir toutes les autres applications.

On envoie le message WM_FONTCHANGE à toutes les fenêtres de niveau supérieur, en utilisant la valeur FFFFH (ou -1) comme handle de fenêtre lors de l'appel à SendMessage. Quand l'application se termine, elle doit supprimer toutes les ressources polices qu'elle a chargées.

```
case WM_DESTROY :
        RemoveFontResource("STOCK.FON");
        SendMessage(0xFFFF,WM_FONTCHANGE,  0, 0L);
```

La fonction RemoveFontResource supprime une ressource police s'il n'y a pas de référence en cours à cette police. Ici encore, le message WM_FONTCHANGE doit être envoyé à toutes les fenêtres de niveau supérieur.

FIGURE 9 : IMPRIMER SOUS DOS

```
FILE *fpPrn;
fpPrn = fopen("prn", "w");
i = 1;
fprint(fpPrn, "%s",
szPrinterInitSequence);
for (iPage = 1;iPage <= nTotalPages;
iPage++)
{
  for (iLine = 1; i <= nTotalLines &&
iLine <= nLinesPerPage; iLine++)
  {
      fprintf(fpPrn, "%s\n",
pszText[i++]);
  }
  fputc(FORMFEED, fpPrn);
}
fprintf(fpPrn, "%s",
szPrinterTerminalSequence);
```

Imprimer sous Windows

L'impression peut se faire au niveau du DOS ou du Bios. Au niveau du DOS, vous employez probablement un petit sous-programme comparable à celui de la figure 9. Ce code peut facilement provoquer une erreur utilisateur, car il ne vérifie pas les conditions d'erreur de l'imprimante : imprimante indisponible, défaut de papier, etc. Si une condition d'erreur est détectée, cela provoque une erreur critique du DOS et votre application est interrompue par un trop célèbre message «Abort, Retry, Fail?» ou «Abandonner, Réessayer, Ignorer?» Ce n'est pas très indiqué pour un programme professionnel.

On peut préférer l'emploi de l'interruption INT 17H du Bios, qui évite le DOS et envoie directement les données à l'imprimante. Lorsqu'on passe par le Bios pour l'impression, l'application doit utiliser un sous-programme pour vérifier les conditions d'erreur de l'imprimante. La plupart des programmes professionnels emploient le Bios afin de protéger l'utilisateur contre de telles erreurs. Tout cela marche très bien si l'utilisateur n'exécute qu'un seul programme. Mais sous Windows, il est possible d'exécuter plusieurs programmes simultanément. Que peut-il se passer lorsque deux programmes (ou plus) tentent d'accéder à l'imprimante? Un programme doit-il se mettre en veilleuse parce qu'un autre est en train d'imprimer? Windows propose un utilitaire d'impression appelé «Gestionnaire d'impression».

Celui-ci agit comme un agent de la circulation et gère l'accès à l'imprimante. Il traite également les conditions d'erreur qui pourraient se produire durant l'impression. Un autre gros problème, pour bien des développeurs, est posé par le support d'imprimante. Il existe des centaines d'imprimantes sur le marché, et la plupart d'entre elles utilisent des codes spécifiques. Un développeur «moyen» doit passer un temps considérable et dépenser beaucoup d'argent pour que son application puisse supporter de nombreux modèles d'imprimantes.

Heureusement, Windows supporte la majorité des imprimantes. En fait, certains constructeurs d'imprimantes proposent eux-mêmes des pilotes d'imprimante à Microsoft pour qu'ils soient intégrés dans l'environnement Windows. Lorsqu'un pilote d'imprimante a été écrit pour Windows, n'importe quelle application Windows peut utiliser cette imprimante, sans programmation supplémentaire (ou très peu). Le contrôle de l'impression est géré, pour sa majeure partie, au niveau du pilote d'imprimante. Quand une application

emploie la GDI pour dessiner sur un contexte de périphérique «imprimante», la GDI stocke les appels dans un métafichier sur le disque dur. Ce métafichier est ensuite envoyé vers l'imprimante. Le pilote d'impression traduit les appels GDI en séquences de contrôle afin de produire l'image sur l'imprimante. Cela requiert plusieurs étapes. La première consiste à obtenir un contexte de périphérique pour l'imprimante. Ce n'est pas aussi simple que de trouver un contexte de périphérique pour une fenêtre. L'application doit déterminer quelles imprimantes sont installées, et lesquelles l'utilisateur veut employer. Le seul moyen de connaître les imprimantes installées est d'examiner le fichier WIN.INI. On trouve les lignes suivantes dans mon fichier WIN.INI :

```
[windows]
        .
        .
        .
device=PCL/HP LaserJet,HPPL,LPT1 :
```

On peut encore lire, un peu plus loin :

```
[devices]
PCL / HP LaserJet=HPPL, LPT1 :
PostScript Printer=PSCRIPT,None
```

L'imprimante spécifiée dans la section [windows] est l'imprimante active au cours des sessions Windows. Les imprimantes spécifiées dans la section [devices] sont celles qui sont potentiellement utilisables. Autrement dit, des pilotes d'impression pour la HP LaserJet et pour les imprimantes PostScript sont installés sur mon disque dur.

On peut utiliser le panneau de configuration pour activer n'importe laquelle de ces imprimantes, ainsi que pour en ajouter d'autres, ou en supprimer. Supposons que l'application veuille utiliser uniquement l'imprimante active. Elle doit lire le fichier WIN.INI et en retirer l'information voulue à l'aide de la fonction GetProfileString (Windows 3.0 fournit également des fonctions privées permettant à l'utilisateur de modifier et de rechercher des fichiers nommés arbitrairement).

Les trois paramètres de l'instruction «device=» sont les trois valeurs nécessaires lors de l'appel à la fonction CreateDC, utilisée pour le contexte de périphérique «imprimante». On peut créer ce contexte avec le code de la figure 10, tiré du programme SDK MULTIPAD.

Maintenant que nous avons un handle pour le contexte de périphérique «imprimante», nous pourrions être tentés d'employer la GDI afin d'expédier les données à l'imprimante. Mais il ne faut pas oublier la séquence d'initialisation envoyée à l'imprimante par la routine d'impression du DOS dans la figure 9! Il faut initialiser l'imprimante, d'une manière ou d'une autre.

FIGURE 10 : CRÉATION D'UN CONTEXTE DE PÉRIPHÉRIQUE «IMPRIMANTE»

```
HDC FAR PASCAL GetPrinterDC(void)
{
    char      szT[32];
    HDC       hdc;
    LPSTR     lpdevmode = NULL;

    iPrinter = 0;

    /* Obtenir l'information de win.ini
dans un tampon et ajouter un zéro
terminal.
     */

    GetProfileString ( "windows",
     "device", "" ,szDevice,
     sizeof(szDevice));
    for (szDriver = szDevice; *szDriver &&
     *szDriver != ','; szDriver++)       ;
    if (*szDriver)
      *szDriver++ = 0;

    /* A partir de la position courante
dans le tampon, ajouter un zéro terminal à
la liste des ports.
     */

    for (szPort = szDriver; *szPort &&
*szPort != ','; szPort++)  ;
    if (*szPort)     *szPort++ = 0;

    /* Traiter les informations
significatives qui peuvent être  contenues
par le périphérique, le pilote et les
tampon des ports.
     */

    if (!*szDevice || !*szDriver ||
!*szPort)
    {
      *szDevice = 0;
      return NULL;
    }

    hdc = CreateDC (szDriver, szDevice,
szPort, lpdevmode);
    return hdc;
}
```

Malheureusement, la GDI Windows ne possède pas de fonction PrinterInitialize. Comme nous l'avons dit précédemment, l'interfaçage entre Windows et l'imprimante est accompli par le pilote d'impression. Windows permet à une application de communiquer

directement avec l'imprimante par des appels à la fonction Escape.

```
nResult = Escape(hPrinterDC,  nEscapeCode,nCount,
                lpInData,  lpOutData);
```

Le paramètre nEscapeCode est une valeur numérique passée au pilote d'imprimante, lui commandant d'accomplir une action quelconque (une liste des codes d'échappement — Escape — est donnée dans le manuel *SDK Reference*, volume 2, chapitre 12). Le paramètre nCount spécifie le nombre d'octets pointés par le paramètre lpInData. Les deux derniers arguments, lpInData et lpOutData, sont des valeurs arbitraires sur 32 bits, dont le contenu dépend du code Escape.

Ils peuvent être parfois des valeurs NULL, des pointeurs sur des chaînes, ou encore des pointeurs sur des structures. Une application doit toujours envoyer le code d'échappement STARTDOC au pilote d'imprimante avant de démarrer l'impression. Cela demande au pilote d'initialiser l'imprimante et l'informe que tous les codes d'échappement NEWFRAME qui suivront doivent être stockés dans le même tampon jusqu'à la réception d'un code d'échappement ENDDOC. Cette méthode empêche qu'un document de plusieurs pages ne soit «mélangé» à un autre document imprimé simultanément. L'appel suivant démarre l'impression :

```
Escape(hPrinterDC, STARTDOC,
strlen(szTitleofDoc),
(LPSTR) szTitleofDoc, NULL);
```

Après le démarrage de l'impression, vous pouvez employer les appels de la GDI Windows pour «dessiner» sur l'imprimante. Par exemple, pour écrire une ligne de texte, le code suivant suffit :

```
TextOut(hPrinterDC,col,row,  "Hello",5);
```

Quand une page entière a été imprimée, l'application doit avancer à la page suivante à l'aide du code d'échappement NEWFRAME.

```
Escape(hPrinterDC,NEWFRAME,0,  NULL, NULL);
```

Cet appel demande au pilote de convertir le contenu du métafichier sur disque. L'imprimante passe également à la page suivante. L'appel NEWFRAME a aussi pour effet de rendre ses valeurs par défaut au contexte de périphérique «imprimante». Cela signifie que si votre application employait une police différente, elle doit resélectionner cette police dans le contexte. Une fois l'impression terminée, l'application doit utiliser le code d'échappement ENDDOC pour informer le pilote d'imprimante que le travail est achevé. Toute tâche d'impression en attente pourra alors commencer.

```
Escape(hPrinterDC,ENDDOC,0,NULL,NULL);
```

Annuler une impression

La majorité des applications Windows qui accomplissent de longues tâches d'impression — tels PageMaker ou Microsoft Word for Windows — permet d'annuler (d'abandonner) l'impression. En règle générale, une boîte de dialogue autorise l'utilisateur à appuyer sur un bouton-poussoir pour arrêter l'impression. Si l'on désire, on peut ajouter une fonction Windows spéciale, appelée «fonction d'annulation» (ou fonction d'abandon).

Une fonction d'annulation est, en gros, une boucle de messages Windows standards. Windows est un système multitâche non prioritaire. D'autres applications Windows peuvent être en cours d'exécution lorsque l'application appelle GetMessage ou PeekMessage. Supposons que l'application se trouve dans une petite boucle, en train d'imprimer le contenu de votre dernier roman. Voici le code de cette boucle :

```
for (iPage = 1; iPage <= nTotalPages; iPage++)
  {
for (iLine = 1; i <= nTotalLines && iLine
              <= nLinesPerPage; iLine++)
  {
  TextOut(hPrinterDC,0,y,szText[i],
              strlen(szText[i]));
  i++;
  }
  Escape(hPrinterDC,NEWFRAME,0,NULL, NULL);
  }
```

Que se passe-t-il s'il ne reste plus de place sur le disque dur pour stocker le métafichier? Le gestionnaire d'impression renvoie une erreur. Mais cela ne signifie pas que l'impression doive s'arrêter — le gestionnaire peut encore imprimer, même si le disque est plein, mais il ne peut pas recevoir de messages.

Le programme est dans une boucle et ne permet pas à d'autres applications d'avoir accès au CPU. Il est donc préférable, et souvent nécessaire, de permettre aux autres applications de traiter leurs messages si vous engagez une longue tâche d'impression. La fonction d'annulation se chargera de cela.

La procédure d'annulation doit être exportée de l'application. Comme pour toute fonction exportée, cela signifie qu'il faut la déclarer dans la section EXPORTS du fichier DEF de l'application, et utiliser MakeProcInstance pour créer un «thunk» instancié. Un

code d'échappement spécial, SETABORTPROC, est utilisé pour informer le pilote d'imprimante de l'existence de la fonction d'annulation.

```
FARPROC lpAbortProc;
lpAbortProc = MakeProcInstance(AbortProc,
hInstance);
Escape(hPrinter, SETABORTPROC, 0, lpAbortProc,
NULL);
        .
        .       // impression
        .
FreeProcInstance(lpAbortProc);
```

La procédure d'annulation peut être rédigée ainsi :

```
int  FAR PASCAL AbortProc (hdc,reserved)
HDC hdc
WORD reserved;
{
MSG msg;
/* Permet à d'autres applications de tourner ou
de recevoir des messages d'annulation */
while (!fAbort && PeekMessage(&msg, NULL, NULL,
NULL, TRUE))
   if (!hwndPDlg ||
!IsDialogMessage(hwndPDlg, &msg))
        {TranslateMessage(&msg);
        DispatchMessage (&msg);
        }
return !fAbort;
}
```

C'est une boucle de messages standards, avec deux ajouts. D'abord, la variable fAbort contrôle le déroulement de la boucle. Sa valeur initiale est FALSE. La procédure de la boîte de dialogue «Abandonner l'impression» met fAbort à TRUE si elle détecte que l'utilisateur a cliqué sur le bouton «Abandon». Voici le code illustrant ce fonctionnement :

```
switch (msg)
{
case WM_COMMAND :
        fAbort = TRUE;
        break; }
```

Il faut encore remarquer, dans la boucle de messages de la fonction d'annulation, qu'il s'agit du code habituel pour traiter les messages des boîtes de dialogue non modales. Il nous faut cela pour nous assurer qu'un clic sur le bouton «Abandon» est bien envoyé à la procédure de la boîte de dialogue. Les informations données ici, ainsi que le code d'impression de notre application de démonstration, devraient satisfaire la plupart des besoins concernant l'impression. La chose la plus importante est qu'il n'est pas nécessaire de se préoccuper des divers modèles d'imprimantes qui existent aujourd'hui. En gérant automatiquement le support d'imprimante, Windows vous fait gagner du temps et vous évite d'écrire des sous-programmes d'impression.

Le programme de démonstration

Le gestionnaire de portefeuille d'actions décrit dans cette série d'articles n'est pas un logiciel professionnel. Il sert simplement d'exemple pour le développement d'une application Windows standard. Certaines caractéristiques de Windows, qui n'ont pas été abordées, peuvent être employées pour améliorer l'application. Nous les évoquerons brièvement pour terminer. DDE (pour Dynamic Data Exchange) est un protocole d'échange de messages permettant à deux ou plusieurs tâches Windows d'échanger des informations. Le DDE peut être employé pour établir un chemin de communication entre un tableur (supportant le DDE) et notre gestionnaire de portefeuille.

Par exemple, après avoir entré les cotations à la fin de la journée, vous pourriez faire une analyse prévisionnelle du portefeuille. Notre application pourrait être utilisée comme serveur de cotations, le tableur étant considéré comme client. Quand le tableur aurait besoin d'une information concernant une action à une date précise, il pourrait envoyer un message DDE à toutes les applications Windows en cours d'exécution, pour demander si une application peut lui fournir cette information.

Si le gestionnaire de portefeuille se trouve en cours d'exécution, il devrait être capable de répondre et d'envoyer les chiffres au tableur. Celui-ci pourrait même être configuré pour lancer une copie du programme STOCK si aucune autre application Windows ne peut lui fournir de réponse.

Il serait encore possible de doter notre programme d'un module de communication. L'API Windows propose des fonctions pour gérer les communications asynchrones. Elles pourraient être ajoutées pour appeler des serveurs télématiques et télécharger des informations boursières.

Rien n'interdit d'imaginer une enveloppe DDE du module de communication pour qu'une autre application puisse amener le programme à se connecter au service télématique et à télécharger les informations afin de les fournir à l'application qui a déclenché le processus. Quant au Presse-papiers de Windows, il est employé pour échanger du texte ou des informations graphiques entre les diverses applications Windows. ❐

SYSTÈME

STRUCTURE DES FICHIERS BMP

Décoder les fichiers BITMAP

JEAN-FRANÇOIS DUVIVIER

Un fichier BMP peut schématiquement se décomposer en quatre parties consécutives. La première, appelée «BITMAPFILEHEADER» fournit des informations générales sur le fichier. La deuxième, «BITMAPINFOHEADER», contient essentiellement des informations sur l'image proprement dite (dimensions, format des couleurs...).

Si les couleurs ne sont pas codées en absolu sur 24 bits (16 millions de couleurs), la troisième partie fait correspondre à chaque code une valeur absolue sur 24 bits. Enfin, la dernière partie, «BITMAP», rassemble le codage complet de chacun des pixels de l'image. La plupart des informations sont stockées sous forme d'un mot de 32 bits (4 octets), afin de mieux coller à l'architecture 32 bits des processeurs 386 et 486, et d'utiliser au mieux leurs registres internes.

L'en-tête du fichier : BITMAPFILEHEADER

Suivant en cela les fichiers EXE et leur célèbre 'MZ', les fichiers BMP sont identifiables par les deux lettres ASCII 'BM' situées dans les deux premiers octets. La taille totale du fichier est ensuite inscrite dans quatre octets, ce qui permet d'atteindre une taille maximale de 4 gigaoctets. Viennent ensuite 4 octets réservés à zéro, puis la valeur de l'offset de la partie BITMAP à partir du début du fichier. Cet en-tête de fichier est toujours de longueur fixe (14 octets).

```
2 octets : bfType       Identification du fichier.
4 octets : bfSize       Longueur totale du fichier.
4 octets : bfReserved   Zone réservée et mise à zéro.
4 octets : bfOffBits    Offset des données bitmap.
```

Les informations sur l'image : BITMAPINFOHEADER

La toute première valeur indique la taille consacrée à la partie BITMAPINFOHEADER. Elle est fixée à 40 octets pour un BMP Windows 3.0 mais peut évoluer dans le futur. Les informations importantes sont ensuite la largeur et la hauteur de l'image en pixels (320 x 240, par exemple, dans PAPIER.BMP livré avec Windows), ainsi que le nombre de bits nécessaires pour représenter un pixel.

Quatre valeurs peuvent être en fait utilisées : 1 pour les images monochromes, 4 pour les images en 16 couleurs (mode EGA ou VGA classique), 8 pour les images en 256 couleurs (mode VGA 320 x 200 ou SuperVGA) et 24 pour les images en couleurs réelles (16 millions de couleurs). L'information indiquant le nombre de surfaces de l'image doit toujours être présente et égale à 1.

Toutes les autres données de BITMAPINFOHEADER peuvent être aisément omises et initialisées à zéro (c'est le cas, par exemple, dans PAPIER.BMP). La taille en octets de l'image est une information redondante car elle peut être calculée à partir des dimensions de l'image et du nombre de bits par pixel.

Les résolutions horizontales et verticales en pixels par mètre donnent la possibilité à une application de choisir parmi plusieurs BMP celui qui conviendra le mieux au périphérique écran employé.

Le nombre de couleurs utilisées et celui de couleurs considérées comme importantes peuvent faciliter le transcodage effectué par un programme d'affichage de fichiers BMP, quand le BMP a une résolution couleur plus importante que le dispositif d'affichage (par exemple, un BMP 256 couleurs sur un écran VGA 16 couleurs). Enfin, il est possible de compresser les données BITMAP dans le cas où le BMP est codé en 16 ou 256 couleurs.

```
4 octets : biSize       Taille occupée par
                        BITMAPINFOHEADER.
4 octets : biWidth      Largeur de l'image exprimée
                        en pixels.
4 octets : biHeight     Hauteur de l'image exprimée
                        en pixels.
2 octets : biPlanes     Nombre de surfaces de
                        l'image (toujours égal à 1).
2 octets : biBitCount   Nombre de bits de couleur
                        (1,4,8 ou 24).
```

Les fichiers BMP sont les fichiers graphiques de base de Windows qui intègrent des primitives pour leur manipulation et leur affichage. Nous nous concentrerons dans cet article sur la structure des fichiers BMP tels qu'ils sont utilisés dans Windows 3.0. Afin de respecter une certaine cohérence, nous utiliserons quelques noms en anglais, tels qu'ils sont définis par Microsoft dans le kit de développement de Windows 3.0.

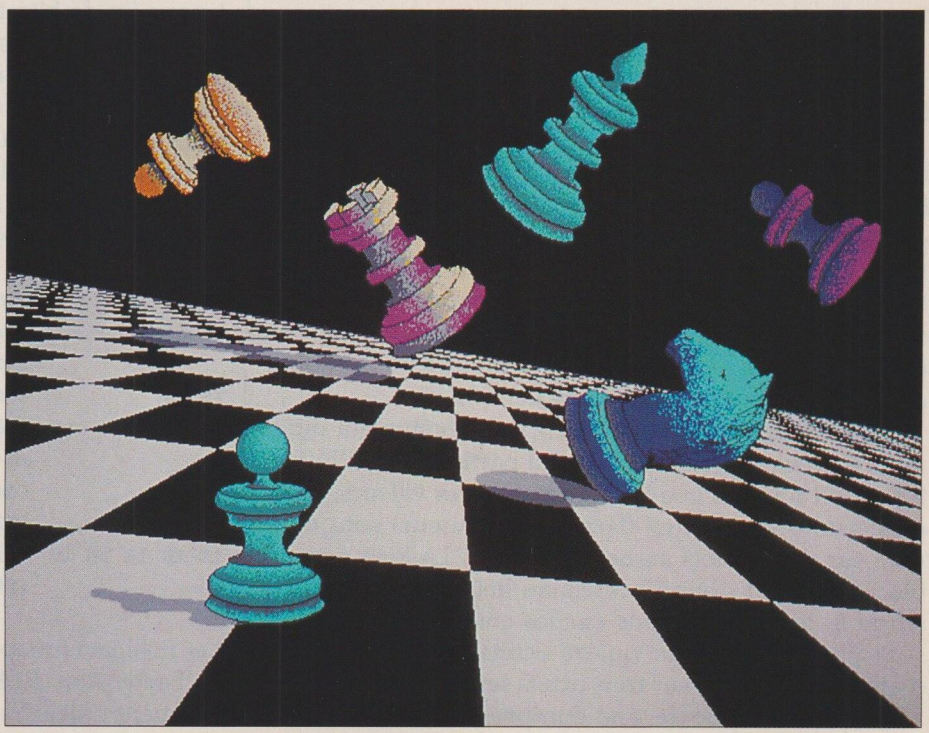
```
4 octets : biCompression Méthode de compression
                         (0=non compressé).
4 octets : biSizeImage   Taille de l'image en octets.
4 octets : biXpelsPerMeter     Résolution horizontale
                               en pixels par mètre.
4 octets : biYPelsPerMeter     Résolution verticale
                         en pixels par mètre.
4 octets : biClrUsed     Nombre de couleurs utilisées
                         (0=toutes).
4 octets : biClrImportantNombre de couleurs
                         importantes (0=toutes).
```

Informations sur les couleurs utilisées

Un codage satisfaisant des couleurs impose 256 niveaux sur chacune des composantes élémentaires RVB (rouge, vert, bleu), ce qui donne 16 millions de couleurs possibles. Seules quelques cartes graphiques haut de gamme donnent accès à toutes ces couleurs. Dans la plupart des cas, une carte graphique permet l'affichage simultané de 16 ou 256 couleurs choisies dans une palette plus importante. Lorsque le BMP est créé sur une carte utilisant une palette, il est nécessaire, afin d'obtenir une image indépendante du périphérique d'affichage, de retraduire les index de couleurs utilisées en «valeurs absolues» codées sur 24 bits.

Quatre octets seront utilisés, car il est plus souple de manipuler des mots de 32 bits. Le premier spécifie le bleu, le deuxième le vert et le troisième le rouge. Le dernier octet doit être laissé à zéro.

Le BMP comprend donc une liste variable de 0, 2, 16 ou 256 couleurs (dépendant du nombre de bits utilisés pour coder chaque pixel), décrites chacune par 4 octets.



Ce jeu d'échecs onirique est livré avec Windows 3.0. Il occupe la totalité de l'écran VGA (640 et 480 points). Le codage des couleurs est réalisé sur 4 bits, ce qui donne 16 couleurs.

STRUCTURE DES FICHIERS BMP

Les deux premiers octets 42 et 40 indiquent qu'il s'agit d'un fichier BITMAP Windows. Les quatre octets suivants donnent la taille du fichier, soit 153 718 octets.

Lorsque les informations biClrUsed et biClrImportant sont employées, la table des couleurs doit être ordonnée en plaçant les couleurs utilisées au début, celles-ci étant classées par ordre d'importance.

Définition des couleurs :

```
1 octet spécifiant le bleu
1 octet spécifiant le vert
1 octet spécifiant le rouge
1 octet réservé à zéro
```

Données BITMAP et méthodes de compression

Chaque pixel de l'image est représenté par 1, 4, 8 ou 24 bits. L'origine est définie comme étant le coin inférieur gauche de l'image, et les pixels sont analysés ligne par ligne. Tous les bits représentant les pixels d'une même ligne sont concaténés entre eux. Cependant, la représentation d'une ligne complète devra, suivant les cas, être complétée avec des bits à zéro afin de toujours occuper un nombre entier de mots de 32 bits.

Les BMP définis sous OS/2 1.x sont légèrement différents de ceux de Windows 3.0. La partie BITMAPINFOHEADER n'inclut que les cinq premières valeurs, et les dimensions de l'image (largeur et hauteur) sont données sur deux octets au lieu de quatre. La longueur du bloc comprend donc 12 octets au lieu de 40 octets. De surcroît, les informations sur les couleurs utilisées ne sont plus données sur quatre octets (RVB et un octet d'alignement) mais sur trois octets seulement.

Windows supporte une compression des formats BMP, si ces derniers sont en 16 couleurs (4 bits par pixel) ou en 256 couleurs (8 bits par pixel). Il utilise un encodage «run length» efficace quand de nombreux pixels consécutifs ont la même couleur. L'idée de base consiste à coder sous forme de deux octets, le premier représentant le nombre de pixels identiques, le second spécifiant leur couleur.

De plus, lorsque le premier octet est mis à zéro, le second indique une action spéciale : 0 pour fin de ligne, 1 pour fin de bitmap, 2 pour indiquer un déplacement relatif décrit par les deux octets suivants et 3 à FFh pour indiquer une séquence absolue non encodée de longueur variant de 3 à 255 octets. Cette séquence doit toujours être alignée sur un nombre d'octets pair.

A titre d'exemple, la séquence : `03 04 00 03 45 56 67 00 02 78 00 02 05 01 02 78 00 00 05 1E 00 01` donnerait les valeurs non compressées suivantes :

```
04 04 04          (trois fois la valeur 04)
45 56 67          (séquence absolue de 3 octets compte
                  non tenu du 00 d'alignement)
78 78             (deux fois la valeur 78)
-                 (déplacement de 5 pixels à droite
                  et d'un vers le bas)
78 78             (deux fois la valeur 78)
-                 (fin de ligne)
1E 1E 1E 1E 1E    (cinq fois la valeur 1E)
-                 (fin du fichier bitmap)
```

Les fichiers bitmap compressés sont généralement dotés de l'extension .RLE (Run Length Encoded) et sont très rarement utilisés. ❐

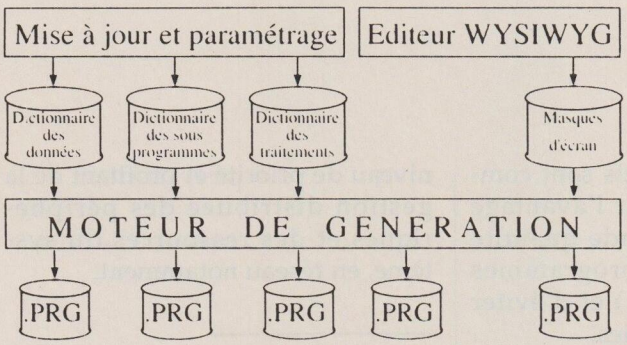
dBGEN 6.1

Générateur d'applications

dBASE III+, dBASE IV, FoxBASE, Nantucket, dBXL, dBFAST.

☛ Une conception à base de dictionnaires

dBGEN est conçu autour de deux dictionnaires : un dictionnaire de données et un dictionnaire de traitement. Ces deux dictionnaires sont complétés par une bibliothèque de sous-programmes.



Une fois les .PRG générés, vous pouvez **modifier le code** spécifique avec votre éditeur habituel, poursuivre vos tests sans regénérer, et en fin de test, **mettre à jour le dictionnaire directement depuis les .PRG** ! Imaginez le gain de temps et de fiabilité !

☛ Un éditeur d'écran WYSIWYG

Le générateur de masques écran donne la **vision réelle** des écrans avec gestion de la couleur. Vous disposez pour réaliser les masques d'un éditeur semi-graphique pleine page avec toutes les fonctions nécessaires à la réalisation de cette tâche fastidieuse dans les temps les plus courts. Que le masque soit fixe ou déroulant, le tracé tout comme le paramétrage du traitement reste le même. La présence du déroulement n'apporte aucune difficulté car **dBGEN génère automatiquement toutes les fonctions de curseur et de pagination**. Le nombre de masques d'une application est illimité.

☛ Un fenêtrage automatique sur fichiers

La gestion des fenêtres est entièrement prise en charge par dBGEN. Il suffit d'indiquer le fichier, son index, les champs à afficher et la position sur l'écran, soit **quelques secondes de paramétrage par fenêtre! Le nombre de fenêtres est illimité.** La gestion de la fenêtre et la sélection sont entièrement automatiques.

☛ Une ergonomie personnalisée

Avec dBGEN vous disposez de **menus déroulants, menus à barre pop-menus,** et fenêtrages. Ces possibilités sont obtenues par des sous-programmes fournis en **code source**. Vous pouvez les modifier ou les remplacer par vos propres sous-programmes et ainsi **personnaliser le "look"** de vos applications. Vous pouvez aussi les intégrer dans une application existante.

Marques déposées : NANTUCKET Corp, dBASE III+ & dBASE IV (ASHTON TATE), FoxBASE (FoxSOFTWARE).

☛ Une gestion simple des relations

La représentation visuelle des relations entre fichiers et la construction assistée de ces relations apportent une grande simplicité dans la définition de l'architecture des programmes. La prise en charge par dBGEN de toute la complexité qui en découle au niveau du code permet au développeur d'en oublier jusqu'à l'existence, et au débutant de réaliser d'entrée des applications multi-fichiers avec écrans déroulants.

☛ Des possibilités étendues de calcul

Vous pouvez réaliser les **calculs les plus complexes** dans vos applications grâce aux très nombreux points d'insertion de programmation ou d'appel de sous-programmes spécifiques. Bien sûr, **vous programmez dans votre langage** avec toutes les fonctions et les appels à vos sous-programmes habituels.

Vous pouvez également **importer des modules de programmes** depuis une bibliothèque, ou cataloguer de nouveaux modules sans sortir de l'éditeur.

☛ Un outil de synthèse et d'analyse

Le survol du dictionnaire vous donne une vue synthétique de l'application avec **représentation graphique** des IF/ELSE, DO WHILE et DO CASE. Pour vous positionner sur le traitement d'un champ, il vous suffit de donner son nom à une commande de recherche automatique. Une à deux secondes plus tard, un ensemble de fenêtres vous donne accès au paramétrage du traitement, à la programmation spécifique et au tracé du masque de saisie. Vous pouvez aussi analyser votre application à partir des masques de saisie.

☛ Un outil de documentation

Le dictionnaire dispose d'une documentation abrégée affichée en permanence à l'écran et d'une documentation étendue de type mémo. A partir de là, dBGEN édite le **dossier de synthèse et de programmation**. Si vous le souhaitez vous pouvez développer votre propre présentation de dossier.

☛ Un outil de portabilité

La description de l'application est **indépendante du langage** et de la gestion du réseau. Ces paramètres sont donnés à la génération. Vous pouvez à tout moment régénérer dans un autre langage, gérer ou non un réseau, prendre en compte ou non **la souris**. Tout cela en fonction des possibilités du langage choisi (dBASE III+, dBASE IV, FOXPRO, NANTUCKET, dBXL, dBFAST).

☛ Les prix

Version Benjamin	850 F HT
Version Junior	3 450 F HT
Version Sénior	9 450 F HT

Présentation les deuxièmes vendredi de chaque mois à 10 H
HOTEL FRANTOUR 163 bis, avenue Clichy PARIS 17
Réservation : ALTI-SOFT 78 72 05 69

Je désire recevoir : une documentation ☐ une version de prise en main ci joint 150 F ☐

Société _____
Adresse _____
_____
Nom _____
Tél. _____

ALTI-SOFT 32, av. Félix FAURE 69007 LYON
Tél. :78 72 05 69 - Fax. : 72 73 00 21

Service lecteurs n° 17 (dernière page)

SYSTÈME

LES OS TEMPS RÉEL SUR PC

Quelle compatibilité pour MS-DOS ?

MAXIME DADDLER

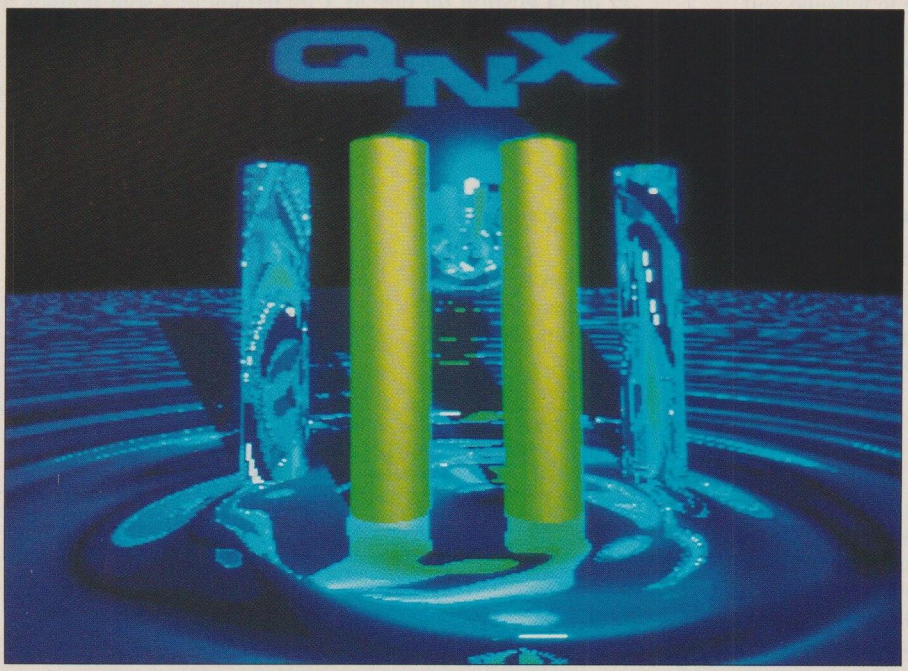
Aujourd'hui, les industriels ne peuvent plus se passer des applicatifs s'exécutant en temps réel! De ce besoin, sont nés il y a quelques années les «exécutifs temps réel». Ces «dispatchers» de tâches ont un rôle bien précis : gérer divers processus en cours d'exécution.

De tels exécutifs se sont rapidement révélés insuffisants pour le développement d'applications en temps réel. Voilà pourquoi on est venu y greffer un certain nombre de tâches et d'accessoires jusqu'à concevoir de véritables systèmes d'exploitation temps réel.

De tels systèmes, s'ils sont compatibles MS-DOS, ont l'avantage de préserver un grande quantité d'investissements (programmes écrits en C, fichiers...) et d'éviter les fastidieuses ressaisies.

Si, de plus, ils permettent d'exécuter MS-DOS, ils donnent la possibilité aux utilisateurs de profiter des applications et des interfaces graphiques liées à DOS, Windows en particulier, sur le même processeur que celui qui fait tourner leurs applications temps réel.

Dans ce cas, certains systèmes vont jusqu'à offrir à l'utilisateur un MS-DOS multitâche, muni d'un niveau de priorité et profitant de la gestion distribuée des périphériques et des ressources du système, en réseau notamment.



Synthèse d'images, animation : rien d'impossible sous QNX.

QNX de Quantum

QNX Software systems ltd (Canada), distribué par Inforange, est un système d'exploitation temps réel et multitâche construit autour d'un micronoyau. Ce dernier constitue un exécutif temps réel très compact dont la taille ne dépasse pas 4 ko. Il gère un ensemble de serveurs appelés «administrateurs» ou «tâches de service» (TSK = TASK). Ces différents serveurs correspondent en réalité à des modules qui peuvent être ajoutés ou enlevés dynamiquement.

Ainsi, DEV gère les liaisons asynchrones, FSYS, les disques, Locker, les conflits d'accès et Network Administrater, le réseau local. Autre principale caractéristique de QNX : son mode de communication entre les différentes tâches. Il s'effectue par l'intermédiaire de messages (Message Passing) que peuvent s'envoyer tous les processus : administrateurs, applicatifs et même micronoyau. Installé sur un PC, QNX peut travailler en mode réel (8086) ou en mode protégé.

Ce dernier lui permet de gérer jusqu'à 16 Mo de mémoire en continu et d'effectuer jusqu'à 255 tâches simultanément. L'adminis-

Les concepteurs de systèmes d'exploitation temps réel sur PC semblent tous avoir opté pour une architecture modulaire bâtie autour d'un micronoyau. Mais la compatibilité avec MS-DOS est encore loin de profiter d'une telle unanimité! Le point sur six systèmes d'exploitation qui offrent de multiples possibilités, de la simple émulation à l'exécution multitâche du DOS avec des niveaux de priorité.

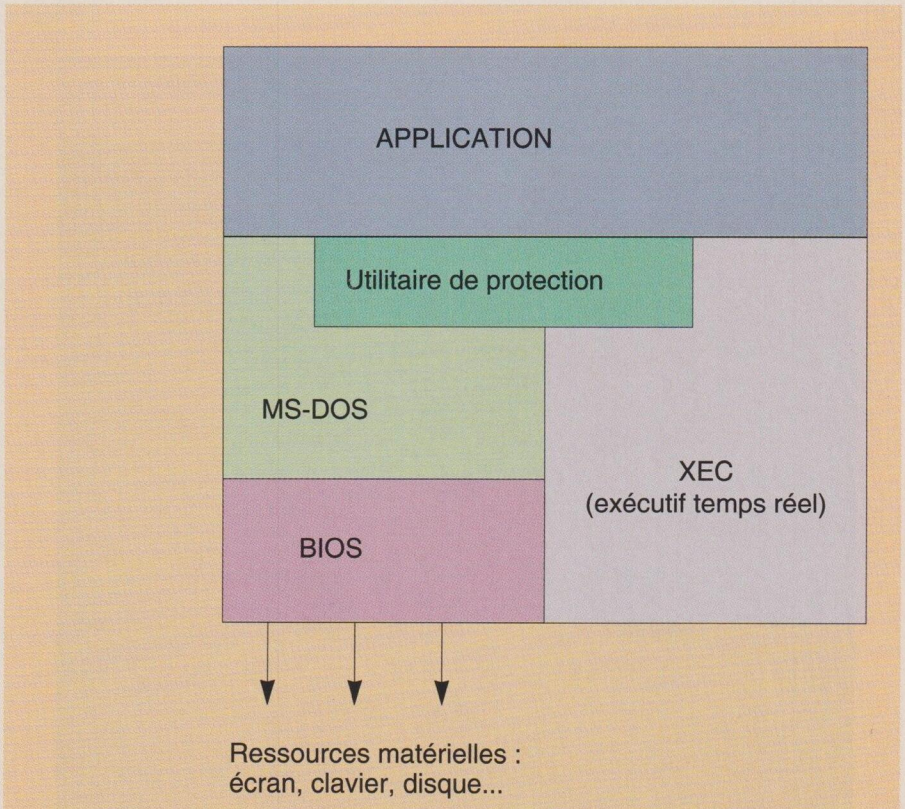
trateur DEV peut alors gérer jusqu'à 32 voies série et 2 ports parallèle. La taille maximale d'une partition est de 2 000 milliards d'octets. Chaque répertoire de la structure de fichiers en arborescence peut contenir 32 000 fichiers, et il est également possible de connecter jusqu'à 50 postes de travail différents sur une même machine.

4 Mo de RAM suffisent

Cependant, QNX nécessite rarement plus de 3 ou 4 Mo de mémoire vive car il fait amplement appel au mode partagé. Dans ce cas, les performances s'en trouvent améliorées.

En effet, plus la taille de la mémoire employée est petite, plus l'utilisateur bénéficie de l'efficacité du cache du processeur. En ce qui concerne la sécurité, chaque utilisateur dispose d'une ou de plusieurs fonctions de cryptage de fichiers ainsi que d'un système de mots de passe et de droits d'accès, fortement inspiré d'Unix.

La répartition des tâches s'effectue, pour sa part, en fonction du niveau de priorité de chacune. Il existe 15 niveaux de priorité référencés sous QNX. Chaque niveau est découpé en tranches de temps (Time Slice), elles-mêmes divisées en tic d'horloge d'une résolution de 1 milliseconde. Le Time Slice, accessible à la programmation, correspond au temps maximal pendant lequel une tâche peut utiliser le processeur.



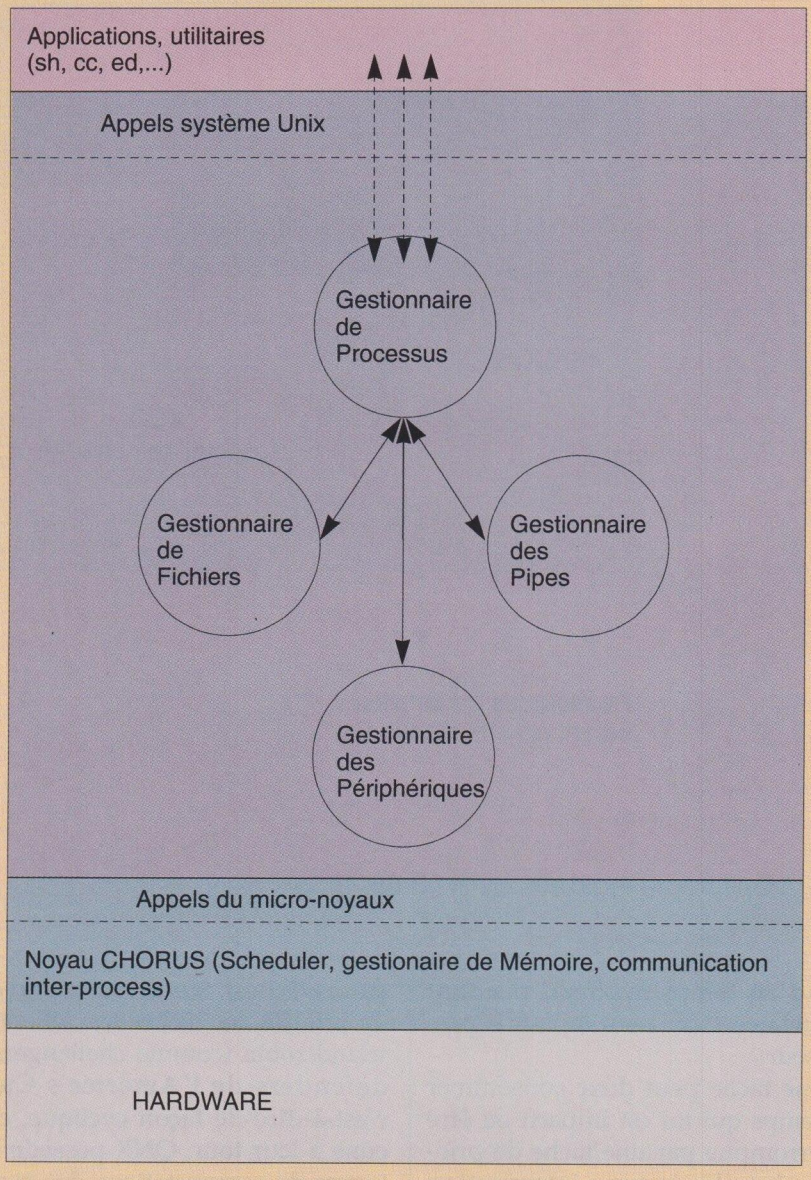
Architecture logicielle de RTC 86

Une tâche peut donc consommer le temps qui lui est imparti ou être interrompue par une tâche de priorité plus élevée (une interruption étant prioritaire sur toutes les autres tâches). Sur un même niveau de priorité, les tâches travaillent en round robin (comme challengers et defenders de l'America's Cup), c'est-à-dire de façon cyclique, chacune à leur tour. QNX possède des temps de commutations des tâches de 50 microsecondes pour un

386/33 et de 20 microsecondes pour un 486/25, alors que son temps de latence moyen varie entre 8 et 25 microsecondes. Ces diverses performances sont conservées si plusieurs PC sous QNX se trouvent reliés par un réseau local.

QNX utilise le réseau local Arcnet de Data Point. Arcnet est un réseau de type Token Pass à 2,5 Mb/s qui assure un débit de 200 ko/s. Il peut atteindre une étendue de 13,5 km si l'on combine câble coaxial et fibre optique. Il est déterministe, c'est-à-dire que l'on connaît à chaque instant les temps de communication. Aucune machine ne s'avère privilégiée sous Arcnet : toutes sont à la fois serveur et client.

Ainsi, n'importe quelle tâche peut avoir accès à toutes les ressources du réseau (périphériques, disques et mêmes processeurs). QNX est un système fermé : l'extérieur ne peut accéder au réseau, strictement réservé au Message Passing.



CHORUS/MIX : l'architecture micro-noyau appliquée au noyau Unix

Par contre, il reste ouvert vers l'extérieur : chaque machine du réseau peut recevoir des cartes de communication (Ethernet, X25, ...) qui procurent autant de ressources globales pour les tâches QNX. Le logiciel DFS de Quantum (DOS File System) assure la compatibilité avec les fichiers DOS sous QNX.

Cet utilitaire permet aux programmes QNX de lire et d'écrire dans les fichiers de toutes les partitions DOS, localement ou à travers le réseau, de façon transparente. Tout fichier peut alors être copié d'une partition DOS vers une partition QNX, et inversement.

Deux versions de QNX

QNX existe en deux versions : QNX standard Ansi v 2.15 et QNX standard Posix v 4.1. Toutes deux partagent la même philosophie (micro-noyau et Message Passing) et diffèrent notamment par leurs compilateurs et par leur interface de programmation.

A noter qu'avec QWindows (une version Quantum de l'Open Look d'AT&T), QNX dispose d'une interface utilisateur conçue pour le temps réel. Son multifenêtrage permet entre autres de voir l'évolution de plusieurs processus simultanément, et en temps réel. De plus, Quantum a développé un éditeur d'interface très pratique sous Open Look. Celui-ci permet d'habiller les programmes de l'utilisateur avec des boîtes à boutons, des menus déroulants, etc., tout cela sans écrire la moindre ligne de code!

Il suffit de sélectionner les différents types de curseurs, boutons, fenêtres et autres dans la boîte à outils d'Interface Editor et de les positionner à la souris : l'éditeur génère alors automatiquement le code correspondant. L'utilisateur peut ensuite insérer sans aucun problèmes ses propres programmes dans les routines existantes.

Au niveau applicatif, QNX dispose avec RUNDOS et QDOS de deux solutions différentes pour obtenir une compatibilité avec MS- DOS. RUNDOS, tout d'abord, est un émulateur DOS. Il s'agit en fait d'une version de MS-DOS 3.3, réécrite pour QNX, qui se charge dans les 640 premiers ko de la mémoire de l'ordinateur. L'utilisateur peut dès lors lancer autant de sessions RUNDOS qu'il y a de machines connectées sur le réseau (une par processeur). Et ce, à partir de n'importe quel poste du réseau Arcnet.

DOS/RMX deviendra bientôt DOS/Windows pour DOS 5.0 et Windows 3.1.

La gestion distribuée en plus

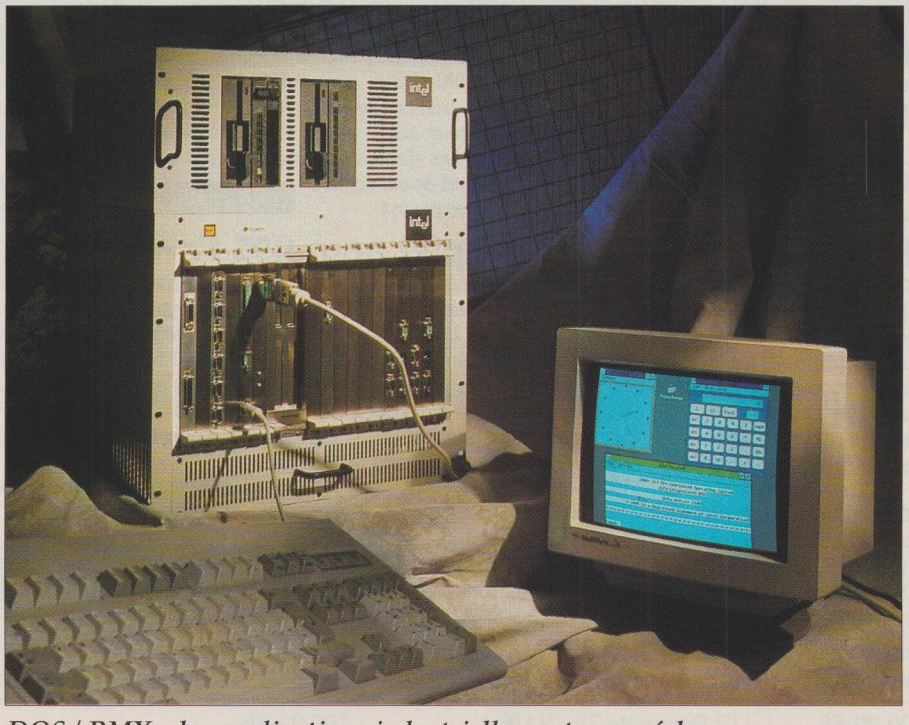
Ainsi, il bénéficie de la gestion distribuée des périphériques et des ressources. Cependant, il ne pourra pas utiliser des applicatifs tels que Windows 3.0 qui attaquent directement les couches basses et le contrôleur. Pourront être utilisés des logiciels tels qu'AutoCAD, Ventura Publisher, Word ou encore Lotus 1-2-3 qui fonctionnent avec une mémoire paginée de type EMS.

Pour travailler à la fois sous DOS et sous QNX, l'ordinateur est obligé de commuter entre le mode réel et le mode protégé (réservé à QNX). Si le passage du mode réel au mode protégé ne prend pas trop de temps, le chemin inverse réclame en revanche quelques millisecondes.

C'est pourquoi on privilégie généralement des applicatifs temps réel au détriment des applications DOS en abaissant le niveau de priorité de ces dernières.

QDOS est un émulateur Bios sous QNX. Si un utilisateur veut lancer ses applications DOS sous QNX, il lui faudra, avant toute chose, installer MS-DOS. Le DOS et ses applications s'exécutent alors comme une tâche de QNX et, par conséquent, profitent de la gestion distribuée des périphériques.

Si QDOS permet aux applications DOS d'accéder au mode protégé, il s'avère malheureusement lourd à gérer et relativement gourmand en temps CPU. C'est pourquoi celui-ci est souvent délaissé au profit de DFS et RUNDOS, demandés par plus de 80% des utilisateurs de QNX.



DOS / RMX : des applications industrielles en temps réel.

DOS/RMX, l'alliance

Le système d'exploitation temps réel DOS/RMX est né du mariage d'iRMX d'Intel et de MS-DOS de Microsoft. Il permet à ses utilisateurs, et aux développeurs, de profiter des applications et des interfaces graphiques liées à DOS, notamment Windows, sur le même processeur que celui qui fait tourner leurs applications temps réel.

A l'amorce du système, MS-DOS

SOYEZ À LA SOURCE POUR VOS DÉVELOPPEMENTS : LIANT

Software, spécialiste des logiciels systèmes pour l'informatique distribuée (connectivité PC, Unix PC), propose aujourd'hui en France les produits de LIANT, la première offre mondiale en termes de langages et outils de développement pour les Systèmes Ouverts :

- LPI C++
- LPI Fortran et RM/Fortran
- LPI PL/1
- CodeWatch debugger source multi-langage avec interface X-Window ou caractère
- LPI C
- LPI Pascal
- LPI Cobol et RM/Cobol
- LPI Basic

Tous ces langages sont conformes aux normes et standards du marché (ANSI, X/Open, GSA, IEEE, FIPS, ISO, SAA, ...) et disponibles sur une **vaste gamme de plates-formes** du PC (DOS, OS/2, Unix), stations de travail aux mini-ordinateurs. Ainsi, il devient possible pour les développeurs de porter leurs applications sur d'autres plates-formes matérielles par simple recompilation.

Software a pour mission en France de distribuer tous les langages de la gamme LIANT en partenariat avec ISE Tech et Planète Graphique qui assurent la distribution respective de C-Scape et Figaro. La Newsletter WatchPoint de LIANT est insérée en encart dans ce numéro.

Pour plus d'informations, contactez Software au :

Tél : (1) 47 61 99 88 - Fax : (1) 49 10 91 04

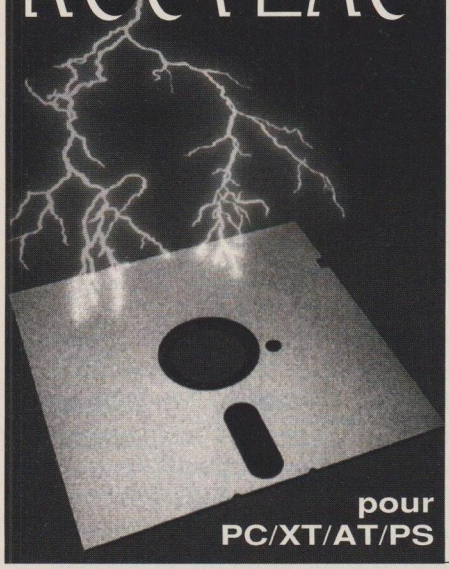
Tour Amboise - 204 Rond Point du Pont de Sèvres - 92516 BOULOGNE Cedex

Software

Service lecteurs n° 18 (dernière page)

Programmez en TURBO-Forth à partir de 100 Fr

NOUVEAU



TURBO-FORTH 83-STANDARD
UN LANGAGE POUR DES APPLICATIONS MULTI-TACHE SOUS DOS

Oui, vous avez bien lu. Pour **100 Fr** (TTC port inclus), vous pouvez utiliser **TURBO-Forth** en version disquette. Cette disquette, disponible au format de votre choix (5'1/4 ou 3'1/2), contient le compilateur **TURBO-Forth** multi-tâche, avec débogueur-traceur, assembleur, décompilateur, l'éditeur ASCII plein écran, tous les fichiers d'auto-documentation intégrée et divers fichiers d'exemple de programmation.

Si vous souhaitez acquérir ultérieurement le package complet, nous remboursons **intégralement** le montant de cette disquette.

Offre valable sans limite de date

TURBO-Forth: package complet pour 995 Fr (TTC port inclus)

TURBO-Forth est la meilleure alternative pour le développement d'applications rapides, compactes, fiables. Particulièrement adapté au contrôle de processus, il est également utilisable pour la gestion du graphisme, le calcul, le traitement de données, les télécommunications...

TURBO-Forth c'est: un interpréteur et un compilateur pour tout système IBM PC/XT/AT/PS ou compatible sous DOS, la programmation structurée comme PASCAL ou C, la définition d'objets nouveaux, un débogueur/traceur en langage évolué, un décompilateur, un assembleur utilisant de véritables mnémoniques, l'exécution de programmes extérieurs à l'application et l'appel à toutes les fonctions DOS, le traitement des nombres dans n'importe quelle base numérique.

La version complète du langage **TURBO-Forth** comprend:
- l'interpréteur/compilateur Forth,
- l'éditeur ASCII plein écran,
- les fichiers d'auto-documentation,
- le méta-générateur Forth,
- les fichiers source de **TURBO-Forth**,
- plus de 40 fichiers sources: gestion port série en langage orienté objet, gestion des fichiers dBASE, virgule flottante CORDIC, animation 3D..
- 2 manuels: **manuel d'apprentissage** et **guide de référence**.

995 FF TTC port inclus

Assistance télématique: 3616 CALVATEL cité FORTH (Hot-line, Forum, Téléchargements). Abonnement gratuit CALVACOM aux acquéreurs de la version complète (permet l'accès via 3614 CALVACOM ou par RTC 2400/4800/9600

☐ TURBO-Forth version disquette	100,00 FF	☐ Disquette TF-CLUB 1	100,00 FF
☐ TURBO-Forth package complet	995,00 FF	☐ Disquette TF-CLUB 2	100,00 FF

commande à adresser à: TVA détaillée sur facture

REM CORP 17, rue de la Lancette 75012 PARIS Tel: (1) 43.42.32.15
fax: (1) 43.40.96.53 MINICOM: 3612 143409653 REM CORP Télétel: 3616 CALVATEL cité FORTH

Service lecteurs n° 19 (dernière page)

WATCHPOINTS

Newsletter for the Customers of Liant Software Corporation Special Edition, May 1992

LIANT TO MARKET C++/VIEWS GUI CLASS LIBRARY

On February 5, 1992, Liant acquired C++/Views, the leading C++ class library for speedily and efficiently developing platform-independent Graphical User Interface (GUI) applications. C++/Views also comes with a powerful browser that simplifies C++ learning and development.

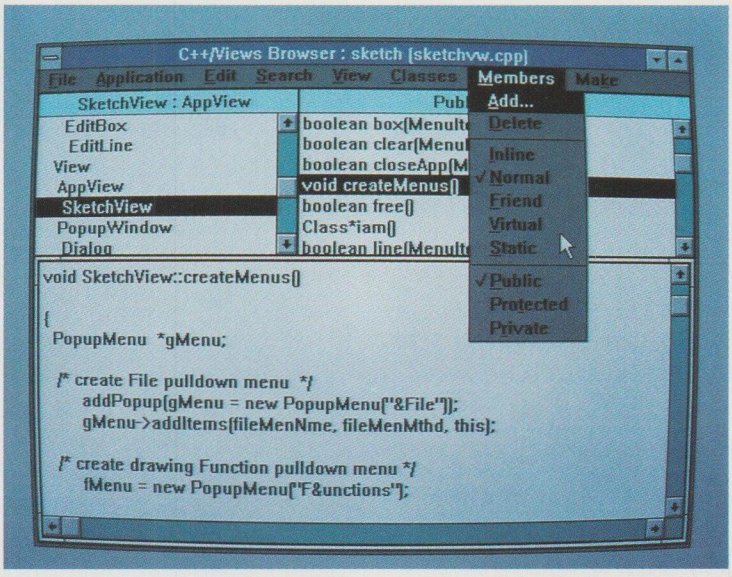
C++/Views, developed by CNS, Inc. of Chanhassen, Minnesota, is used by more than 2,500 customers and offers a rich set of 100 classes that support GUI development. It includes foundation classes for ordered collections, strings, file management, memory allocation, stacks, dynamic arrays, associations, and much more. These foundation classes are invaluable for character-based applications as well.

According to Stephen M. Kuhn, director of marketing at Liant, "By eliminating the need for the developer to deal directly with the underlying GUI manager, such as Microsoft Windows or X Windows, C++/Views provides more than platform independence — it provides the optimal method for extremely rapid GUI development."

Jim Schwarz, lead architect and developer of C++/Views, said, "We are very excited. Liant's commitment to quality C++ development tools is well known, and C++/Views demonstrates that Liant will make a significant contribution to that continuing effort."

"C++/Views makes getting into C++ much easier," added Kuhn. "Its source-code browser is a tremendous learning tool for novice C++

(Continued on page 3)



Liant's C++/Views Browser.

LIANT LAUNCHES LPI-C++ WITH CODEWATCH DEBUGGER SUPPORT

On September 17, 1991, Liant announced the introduction of "LPI-C++", a true 32-bit compiler for application development on multiple UNIX platforms. Packaged with "CodeWatch", Liant's full source-level debugger, Liant's newest product provides UNIX programmers with a time-saving, easy-to-use solution for building reliable, portable applications with C++.

According to Roy A. Finney, president and CEO of Liant, "We believe LPI-C++ is the most powerful C++ compiler available for UNIX systems. The availability of LPI-C++ is a milestone for Liant, because object-oriented technology is fundamental to meeting our customers' needs in the 90's. C++ is the cornerstone of our future strategy."

LPI-C++ is a complete implementation of C++ as described in the definitive *Annotated C++ Reference Manual* by Ellis and Stroustrup. LPI-C++ also has compile-time switch compatibility with "cfront" (1.2, 2.0, and 2.1), ANSI C and Classic (K&R) C.

LPI-C++ includes Liant's new optimizer. It greatly reduces program size and improves run-time performance using such features as global common subexpression elimination, branch chaining, loop induction, loop unrolling and subroutine in-lining.

(continued on page 2)

IN THIS ISSUE

1 C++/VIEWS
LPI-C++ WITH CODEWATCH

2 LPI-C++ (CONT.)
FROM THE CEO

3 C++/VIEWS (CONT.)
SOFTWAY MARKETS LIANT PRODUCTS
RM/PANELS V1.1

4 C-SCAPE APPLICATIONS
NEW X SUPPORT FOR C-SCAPE

5 LIANT ADS

6 RM/PANELS SCREEN MANAGER
PER-USER PRICING — RM/COBOL

7 RM/TECHNOTES
C-SCAPE 3.2C VMS

8 RM/PLUS DB LINKS COBOL WITH 4GLS
RM/COMPANION RELEASED

LIANT'S LPI-C++ WITH DEBUGGER SUPPORT

(continued from page 1)

LPI-C++ gives programmers time-saving options when developing and debugging applications. The debugger allows programmers to enter and display all names as they appear in the C++ source code; also to evaluate C++ expressions and monitor objects in the debugger using C++ syntax. The debugger not only offers many traditional debugging facilities for C++, but also offers powerful class browsing, data browsing, and easy access to overloaded functions.

"LPI-C++ increases productivity, because programmers are able to debug their applications using the symbols and conventions of the C++ language — and they never have to deal with mangled names," according to Don Dudley, Liant's vice president of UNIX development.

Now available from Liant on Intel i386, i486-based systems running UNIX V.3 and on SPARC systems running SunOS, LPI-C++ will soon be available on Intel-based systems running UNIX V.4. Users can interact with the debugger using the X Windows and Motif graphical interface or a character-based interface. ▲

Roy Finney, CEO of Liant Software Corporation

From the CEO...

This issue of watchpoints exhibits the new design we are implementing for all Liant communications. This theme is carried throughout our new advertising, which you may have seen in Dr. Dobb's Journal, Computer Graphics World, UNIX Today!, BYTE, and numerous other publications. We have also recently completed the new Liant product catalog, "The Source". If you would like to receive one, please call +(33) 1476-199-88.

The changes go deeper than design and graphics, however. The new LPI C++ compiler with debugger support has been released, emphasizing our focus on object-oriented technology. New versions of C-scape and RM/PANELS are also shipping.

We are also expanding sales efforts in Europe (as well as in the U.S.).

Please let us know your suggestions or comments, by telephone or fax, on how we might serve you better.

Roy A Finney

LIANT TO MARKET C++/VIEWS CLASS LIBRARY

(Continued from page 1)

programmers. And experienced developers will really appreciate its navigational capabilities."

Rounding out C++/Views' extensive features is complete support for Microsoft Windows Dynamic Link Libraries (DLLs), Dynamic Data Exchange (DDE), and Multiple Document Interface (MDI).

A Success Story

PeoplePro Inc. of Emeryville, California, uses C++/Views to create a sophisticated management productivity tool targeted at Microsoft Windows and the Macintosh.

"Our application relies heavily on the graphics and drawing features of GUI environments, and the implementation of this functionality in C++/Views is powerful and intuitive," said Gerald B. Huff, vice president of product development. "It's a very well-designed and well-implemented library that very effectively shields you from the complexities of GUI programming."

Huff estimates that they have shaved 40% off of their development time by using C++/Views. "It's a tremendous timesaver because it comes with predefined classes that model all the controls and behavior of GUI environments. The foundation classes have also been indispensable."

Liant formally announced C++/Views at Software Development '92 in the U.S, on February 25-27.

C++/Views is fully compatible not only with Liant's C++ compiler but also with a number of other leading C++ compilers including Borland C++, Microsoft C V7, and Zortech C++.

C++/Views supports GUI development for Microsoft Windows. Versions for UNIX X Windows (Motif) and OS/2 Presentation Manager are in beta testing now and will be available in the second quarter of this year.

For more information or a free brochure on the product, please contact Liant in Europe, at our London office at +44 (0) 71-799-2434 or in Berlin at +49 (0) 30-391-5045. In the U.S. contact us at 1-(800)-662-9866.▲

SOFTWAY TO MARKET LIANT PRODUCTS IN FRANCE!

Softway, expert in system software for distributed data processing (connectivity between PC and Unix) is now the exclusive French distributror of Liant, the first source in terms of languages and developement tools for Open Systems :

- LPI C++
- LPI C
- LPI Fortran and RM/Fortran
- LPI Pascal
- LPI PL/1
- LPI Cobol and RM/Cobol
- LPI Basic
- CodeWatch Debugger multi-language source debugger with X-Window or character interface.

All these languages follow the market's specifications and standards (ANSI, X/Open, GSA, IEEE, FIPS, ISO, SAA,...) and are available on multiple platforms (DOS, OS/2, Unix) from PCs to workstations and mid-frames. It is thus possible for developers to transfer their applications from one platform to another by simply recompiling it.

Softway will distribute in France the whole line of LIANT languages. For further information please contact Softway at:

Tel: (33) 1 47 67 99 88

Fax: (33) 1 49 10 91 04

FOR LIANT PRODUCT INFORMATION, CALL FOR OUR CATALOG THE SOURCE (33) 1 47-61-99-88

RM/PANELS V1.1 BEGINS SHIPPING

We have updated RM/PANELS to provide more flexibility to application users. In RM/PANELS version 1.1, a new end-user editor permits the user to access customization and configuration options by typing a single "hotkey". Application-builders using RM/PANELS are able to define screen layouts, color and video attributes, help and error messages, and entry validation outside of the COBOL application.

With RM/PANELS 1.1, the application user can modify default values and attributes without changing the underlying panel definition or the COBOL application.

Additionally, RM/PANELS 1.1 now supports line drawing capabilities in the panel layout editor. Version 1.1 also implements extended character-level validation capabilities for fields.

RM/PANELS is available on all platforms running version 5.0 or higher of RM/COBOL-85.▲

APPLICATIONS BUILT WITH LIANT'S C-SCAPE

Liant's C-scape Interface Management System is a library of C functions for easily creating professional user interfaces. Since C-scape's initial release in 1986, 10,000 programmers around the world have used it to create professional applications for a variety of uses. Some are used within a company to increase the competitive advantage, while others are sold commercially. Here are some commercial applications that use C-scape for their user interface.

Media Group/Quick Tally Systems
Quick Tally's interactive network is an instantaneous feedback and information gathering system that propels the science of audience response measurement into the electronic age. Participants — the audience for America's Funniest Home Videos, for example — enter their responses into a handset resembling a TV remote control unit.

The Quick Tally software collects the data, tabulates it, and instantly displays the group's response. An immediate cross-tabulation function can break out information on any subgroup. For more information on Media Group/ Quick Tally call 212-582-8050.

Postalsoft, Inc.
Postalsoft's ECA OCR/CS Secondary Sort Retrofit Kit saves the United States Postal Service millions of dollars in labor costs every year. It reads multi-line addresses, performs optical character processing, parses address lines, and cross-checks against a computerized directory; it then returns a carrier route code to allow automatic stacking and bar code spraying directly onto the mail pieces for subsequent sorting. For more information on Postalsoft, call 608-788-8700.

Personics
Monarch is Personics' data extractor that gives you instant access to data buried deep in computer-generated reports. Instead of spending hours flipping pages and typing numbers into a spread-sheet, you can use Monarch to read the report in ASCII form, filter information according to your specified criteria, perform any calculations, and output the informa-tion to your spread-sheet, word processor, or
database manager. For more information on Monarch, contact Personics at 508-897-1575 or 1-800-445-3311.

ESM
ESM's DataDesigner is a programming tool that combines C-scape and Raima Corp's db_VISTA into a complete application generator. DataDesigner lets you retain all the power, flexibility and speed of the C language. And its open architecture allows you to link in your own code as well as other C libraries, such as graphics, communications, or mathematics packages.

DataDesigner offers WYSIWYG screen design, SAA-compatible menus, high-speed DBMS, and portability between DOS, QNX, UNIX, and VMS. For more information on DataDesigner, call Liant at 1-800-662-9866; or call ESM in Europe at +49-70-2234-07.

For more information on C-scape and Look & Feel, contact Liant Software Corporation at 1-800-662-9866.▲

NEW! — X SUPPORT FOR C-SCAPE 3.2C UNIX

Liant Software Corporation recently announced version 3.2c of C-scape UNIX, which includes support for the X Window system. C-scape is a comprehensive library of C routines for creating professional user interfaces. The new X support will give C-scape developers the ability to easily write applications for X that include data entry, text editing, menus, help, graphics, and mouse support.

Existing C-scape applications can also be recompiled under X. C-scape is a portable C programming library that helps both novice and experienced programmers design their user interfaces and easily create GUI applications. To move any C-scape application between supported operating systems, users need only recompile for the system they want to use.

"Our X support will offer a simple way for developers to create sophisticated applications for X," said Joe DeSantis, vice president of Liant. "Since C-scape shields the user from operating system specifics, you can write applications for X without having to do any X programming yourself."

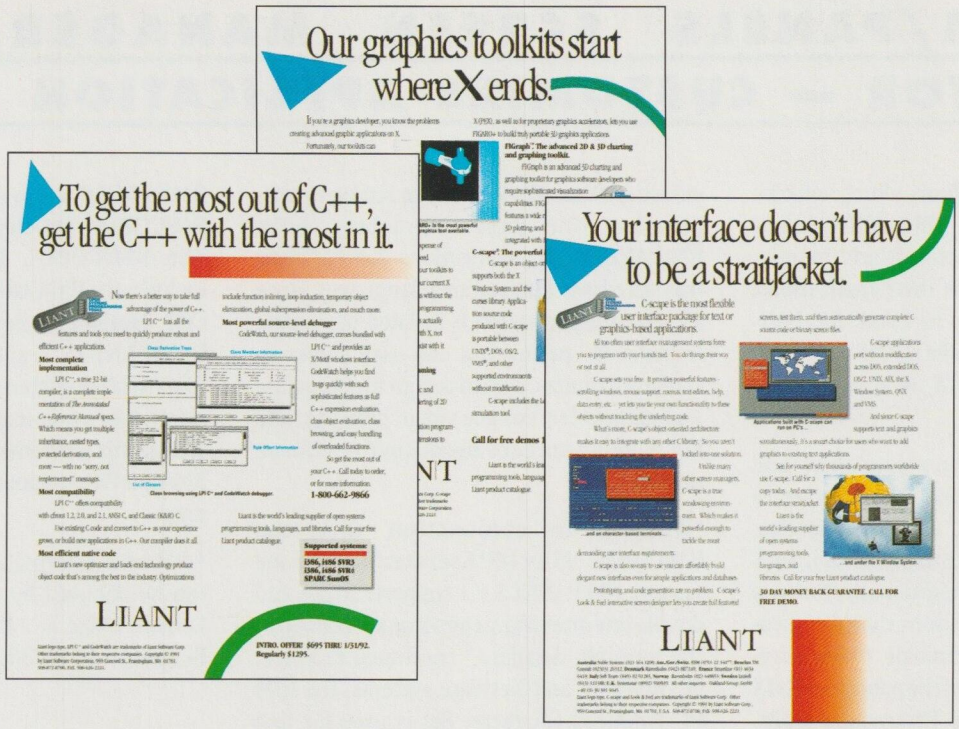
C-scape comes with the Look & Feel Screen Designer. Users can prototype and simulate screens, and automatically turn them into C source code that runs under UNIX, VMS, MS-DOS, OS/2, and QNX. Look & Feel allows users to create screens with all of C-scape's features, even if they have no programming experience. C-scape applications can include sophisticated features, such as menuing systems, borders, and pop-up windows. Text editing functions include word wrap, search and replace, and block commands. The mouse support allows users to manipulate windows or select menu choices.

Look & Feel users can create data entry, main menu, and title screens with the keyboard or the mouse. Users can lay out text, draw lines, choose colors, and select other functional and display attributes. The straightforward interface makes it easy to go from an idea to a functional prototype in a fraction of the time usually required — even for non-technical personnel.

First shipped in 1986, C-scape is now used by major corporations, government agencies, and consultants in over 40 countries. C-scape is shipped with a set of example programs with source code that users can use or modify for their own needs. A user's manual, a function reference, and a quick reference card are also included.

In addition to C-scape and Look & Feel for UNIX, versions for VMS, DOS, OS/2, and QNX are also available.

Customers have access to a 24-hour bulletin board. In the U.S., a demonstration kit (including DOS demonstration disks) is available by calling Liant at 1-800-662-9866. In Europe, customers should contact their C-scape dealer or distributor or Liant's Berlin office at +49 (0) 30-391-5045.▲

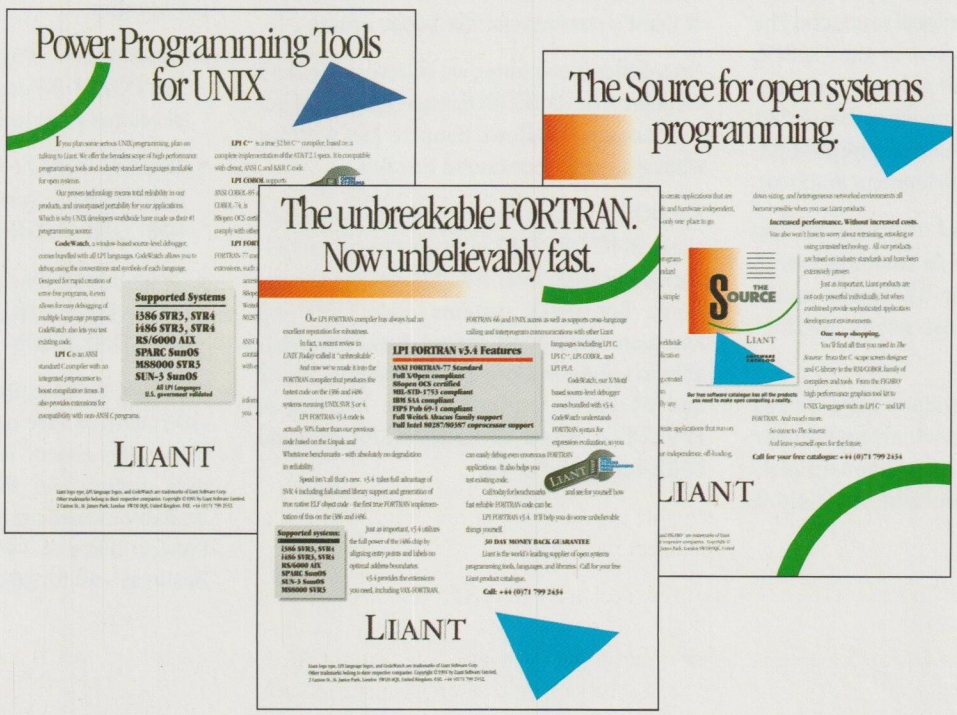


LOOK FOR OUR NEW LIANT ADS

Liant's new advertising features a common "look and feel" for all products. Shown here are some of the ads we are currently running in trade publications.

Liant products advertised include C-scape, LPI FORTRAN, Power Programming Tools for UNIX, LPI C++, and shortly, FIGARO+ graphics and RM/COBOL product families.

Another dimension to our new "look and feel" is the introduction of new sales collateral, technical documentation, and packaging. For example, this issue of *watchpoints*!▲



NEW RM/PANELS SCREEN MANAGER/FORMS GENERATOR — CUSTOMIZE APPLICATION SCREENS

A new panel editor, which enables end-users of RM/COBOL-based commercial applications to customize the appearance of their applications, was announced recently by Liant. RM/PANELS version 1.1 is available for MS-DOS, OS/2, Novell NetWare, UNIX, AIX, and UNIX-compatible systems.

"Both COBOL application developers and end-users will enjoy the new features of RM/PANELS version 1.1," said Mike Prentice, vice president of customer relations for Liant's commercial products. Additionally, the new version supports the line draw capability provided by a variety of terminals, thereby improving the appearance of the panels on UNIX and UNIX-based systems. This capability provides new flexibility in creating screens incorporating line draw characters. With RM/PANELS 1.1, applications can easily be translated into international languages, since all textual information is isolated in a single module. By permitting end-user access to screen attributes, screen design functionality including color, reverse video, blinking, and highlighting can be controlled by the ultimate user.

"Working in conjunction with the RM/COBOL-85 for UNIX version 5.1 run-time system, RM/PANELS 1.1 supports single and double line draw characters that are defined in the terminfo database," continued Prentice. On MS-DOS and NetWare systems, the IBM PC Extended Character Set is used for line drawing.

RM/PANELS 1.1 is available as part of the RM/COBOL Developer Pack Plus bundle. The RM/COBOL Developer Pack Plus includes the RM/COBOL-85 compiler, the RM/CO* developers' environment, and the RM/COMPANION report writer. Existing RM/PANELS customers can upgrade to version 1.1 for a nominal fee. There is no upgrade charge for customers participating in the RM/CareWare support and maintenance program.

Additional information on RM/PANELS 1.1 can be obtained by contacting Liant at our London office at +44 (0) 71-799-2434 or in Berlin at +49 (0) 30-391-5045.▲

NEW PER-USER PRICING FOR RM/COBOL PRODUCTS

Responding to customer requests and emerging trends in the software industry, Liant Software Corporation has announced a new per-user pricing strategy for its RM/COBOL family of products. The new price options — a first in the COBOL compiler market — will allow Liant's customers to save money by paying for software that is based on the number of users of the product, rather than the type of development platform.

The company introduced the new pricing structure in order to give its customers more options that reflect their day-to-day needs, and announced new cost-saving product bundles.

"Until now, COBOL vendors have priced products not according to actual programmer usage, but according to a development platform — an expensive and inefficient approach from a customer's perspective. We found that our customers prefer pricing that reflects the actual number of users of a product," said John Bradley, senior vice president of Liant's commercial ISV buyer group.

"In a difficult economy, we believe customers will welcome RM/COBOL's new pricing structure because it allows them to pay only for what they use," continued Bradley.

For DOS-based platforms, customers can purchase RM/COBOL-85 in five versions, including single-user, five-, 10-, 20- or 50-user options. For UNIX systems, RM/COBOL-85 and associated development tools are available in configurations for four, eight, 16, 32 and 64 users. Unlimited user licenses of RM/COBOL-85 are also available for DOS and UNIX-based platforms.

New Product Bundles

In addition, Liant announced the immediate availability of several new RM/COBOL product bundles:

- Developer Pack, including the RM/COBOL-85 compiler and the RM/CO* developers' environment.
- Developer Pack Plus, which adds the RM/PANELS screen manager and the RM/COMPANION report writer.
- Deployment Pack Plus, which bundles both RM/COBOL-85 run-times and RM/COMPANION run-times.

Purchase of these product bundles will allow customers to save up to 40 percent on individual component list prices. Both the per-user pricing and the product bundles are available immediately. Please call Liant at our London office at +44 (0) 71-799-2434 or in Berlin at +49 (0) 30-391-5045.▲

RM/TECHNOTES REVISED & UPDATED

In December, an all-new RM/Technotes was sent out to registered users of RM/COBOL products. RM/Technotes is a quarterly newsletter published by the RM/COBOL customer support group, and contains technical information of interest to all users of RM/COBOL products.

Along with an updated look, the new RM/Technotes is dedicated to providing timely information about new and better ways to use RM/COBOL and related products. Mike Prentice, vice president of customer relations, says: "We want RM/Technotes to be the standard method of keeping our users up to date on new products, tips and tricks, and ways to get the most out of their RM/COBOL products."

If you aren't currently receiving RM/Technotes and would like to, contact your Liant sales representative.▲

RM/Technotes

Fall 1991 Volume 2, Number 3

RM/COMPANION 1.0

RM/COMPANION is the new RM/COBOL information retrieval and report generation system. Its fourth-generation features make it an ideal system for both developers and non-technical end-users. Because it was created to work specifically with RM/COBOL data files, RM/COMPANION runs faster and with lower overhead than other report writers.

The powerful, English-like command language supports sorting, selection, calculation and formatting. Your existing RM/COBOL data files can be quickly and easily searched, manipulated, and printed or displayed according to your specifications. The data gathered by RM/COMPANION can later be reformatted, graphed and plotted.

RM/COMPANION provides the option of generating reports in either compiled or interpretive mode for maximum flexibility. By selecting the compiled option, developers can generate an RM/COBOL-85 object program which can then be distributed to end-users without the RM/COMPANION runtime component. If more end-user control is desired, the interpreted reports can be distributed with the end-user version, allowing for report customization and new reports from the application's data dictionary.

RM/COMPANION is available on all platforms running version 5.0 or higher of RM/COBOL-85.

RM/PANELS 1.1

RM/PANELS has been updated to provide more flexibility to application users. A new end-user editor permits the user to access customization and configuration options by typing a single "hot key." Application-builders utilizing RM/PANELS are able to define screen layouts, color and video attributes, help and error messages, and entry validation outside of the COBOL application. With RM/PANELS 1.1, the application user can modify default values and attributes without changing the underlying panel definition or the COBOL application.

Additionally, RM/PANELS 1.1 now supports line drawing capabilities in the panel layout editor. Version 1.1 also implements extended character-level validation capabilities for defined fields.

RM/PANELS is available on all platforms running version 5.0 or higher of RM/COBOL-85.

CONVERTING RM/COBOL-85 FILES TO BTRIEVE FILES

Many of you are taking advantage of our NetWare product to allow your applications to run with Novell's Btrieve file system. One question we often hear is "How do I convert my files?". RM files may be converted to Btrieve files in one of three ways.

First, we provide a conversion utility with the NetWare product that converts an RM/COBOL-85 indexed file to Btrieve without having an external file definition. Our utility uses Novell's RMSX.BTRV program during the conversion run. This program is included with Novell's development kit.

Second, RM/COBOL-85 indexed files can be converted to a Btrieve file using a very simple COBOL program. Starting the Btrieve adapter with CREATE=YES will cause new files to be written to Btrieve. A program that opens the existing RM/COBOL-85 indexed file and retrieve new output file will convert the file. The RMSX.BTRV program from Novell is not required for this conversion. However, the file definitions must be available to write the COBOL program.

Third, many commercial applications provide a utility to reload or rebuild indexed files. Again, setting the CREATE=YES option on the Btrieve adapter prior to running one of these programs will convert the file.

Converting Btrieve files is simple. Running existing applications with Btrieve is simpler yet. Once the file is converted, the RM/COBOL-85 runtime system automatically looks for the Btrieve adapter and finds the converted files. No special Novell utilities or options (other than Btrieve itself) are required.

NON-DEDICATED SERVER FIX AVAILABLE FOR NOVELL 2.2

Customers upgrading to Novell 2.2 on a non-dedicated server are unable to run RM/COBOL-85 without suffering an abend.

This problem was referred to Novell and a patch is now available. Please contact your Novell representative and ask for 22NDZF.ZIP. This patch is also available on Compuserve.

ADDITIONAL CONVERSION TOOLS NOW AVAILABLE

In the April 1991 issue of *RM/Technotes*, a conversion program for Micro Focus COBOL/2 files was announced. A new conversion program is now available to allow customers to convert Acucobol indexed files to RM/COBOL-85 indexed files. If you would like more information on these conversion programs, please contact customer support at 1-513-343-1010.

NOW YOU CAN DO IT FROM COBOL

Now You Can Do It From COBOL is a subroutine library developed by Rogers Systems, Inc. which is now available directly from Liant. The library contains over 35 communications, I/O and system subroutines for DOS operating systems. With *Now You Can Do It From COBOL*, you can perform many functions traditionally unavailable from COBOL.

For more information, please contact your RM/COBOL sales representative.

FREEING UP MEMORY ON DOS APPLICATIONS

The RM/COBOL 5.10 runtime for DOS allocates a larger amount of memory to sort routines. The default is now 256K bytes; the previous default was 16K. If you do not use a sort, or are having memory problems with your routines, use the runtime parameter "T=n" to allocate less memory to the sort. A setting of "T=16000" restores the sort allocation to the previous default. If no SORT or MERGE is used, a setting of "T=0" may be specified.

In addition to the change in the default sort memory, the DOS RM/COBOL 5.10 runtime also uses a larger number of file handles. By using the SHARE command parameters, you can allocate additional file handles, etc.

NOTE: DOS 4.0x will load SHARE automatically if you have partitions of 32mg or greater. In order to set your own SHARE parameters, it is necessary to prevent DOS from having access to SHARE.COM. This can be done by renaming it. The renamed file (SHARE1.COM) can then be loaded with the parameters you specify from the autoexec.bat file.

Liant's updated RM/Technotes.

LIANT ANNOUNCES C-SCAPE 3.2C VMS

Liant Software Corporation announced the availability of C-scape 3.2C VMS, a new version of the C-scape Interface Management System that gives VAX/VMS programmers the flexibility to develop sophisticated graphics- or character-based applications. A powerful, object-oriented interface management system, C-scape 3.2C VMS is a comprehensive library of C routines for creating professional user interfaces.

The flexibility to design graphics- and text-based applications gives C-scape VMS programmers the freedom to develop applications in any VMS-based computing environment, according to Joseph DeSantis, vice president of products and technology.

"With C-scape VMS, programmers can now create sophisticated applications that will run on character-based terminals or the most advanced graphics-based platforms, because C-scape supports SMG (DEC's VMS Screen Management Facility) and DECwindows simultaneously," said DeSantis. "This capability not only gives programmers flexibility, but it also protects the user's investment in hardware development platforms."

C-scape VMS runs in graphics mode under DECwindows and in text mode by calling SMG. C-scape VMS under DECwindows uses Xlib to gain full color graphics and mouse support. Under DECwindows, C-scape programmers can load .pcx files into windows and specify X icons. In text mode, C-scape VMS runs on all SMG terminals including the VT52, VT100, VT200 and VT300 series. To run current SMG-based C-scape VMS under DECwindows, users simply recompile. Users can also port C-scape VMS programs to other operating platforms easily, according to DeSantis.

"Liant allows programmers to design enduring applications by ensuring that users can port them easily among heterogeneous computing platforms. All of the hardware-dependent functions in C-scape VMS are encapsulated to allow users to port programs from VMS to operating environments such as DOS, extended DOS, OS/2, QNX and UNIX," DeSantis said.

A full-featured windowing environment, C-scape VMS allows programmers to create an unlimited number of interactive windows, including popup, tiled, overlaid and exploding windows. Users can scroll windows and modify borders with titles, prompts, scroll bars and the mouse. Users also can control the behavior of fields, as well as routines for creating custom field functions.

C-scape VMS comes with "Look & Feel", Liant's screen designer that allows programmers to prototype and simulate screens and automatically turn them into C source code that runs under VMS, UNIX, MS-DOS, OS/2 and QNX. C-scape VMS and Look & Feel supports VAX C v.3.0 or higher and integrates with other C tools and C libraries.

For more information on C-scape VMS and Look & Feel, contact Liant at 1-800-662-9866.▲

Liant Software Corporation
corporate headquarters:
Liant Software Corporation
959 Concord Street
Framingham, MA 01701
(800) 662-9866
(508) 872-8700

Europe:
Liant Software Ltd.
2 Caxton Street
London SW1H0QE
United Kingdom
(44) 71-799-2434

Germany:
Oakland Group GmbH
Alt-Moabit 91 B
W-1000 Berlin 21
Germany
(49) 30-391-5045

Japan:
Nippon Liant
31-8, Takasecho
Funabashi City
Chiba 273, Japan
(81) 47-437-9816

Contributors to this issue:
Jeff Burnett, Stephen Kuhn, Steve Murchie, Carol Pollard, and April Seaberg.
For *watchpoints* information, contact April Seaberg.

Liant Software Corporation offers the broadest scope of programming languages and development tools in the open systems marketplace. Liant provides developers with a choice of languages, environments, graphics and user interface tools to develop, maintain and deploy applications that are consistent across the full range of platforms and operating systems from PCs to supercomputers.

watchpoints is the quarterly customer newsletter of Liant Software Corporation. All rights reserved. The companies mentioned herein own various trademarks. The Liant logo type, C++/Views, C-cell, C-spot, CoEdit, FIGARO+, FIGraph, LPI-language logos, PCI, PSO, RM/COMPANION, RM/FORTRAN, RM/PANELS, and RM/plusDB are trademarks, and CodeWatch, C-scape, Look & Feel, RM/COBOL, RM/COBOL-85, and RM/CO* are registered trademarks of Liant Software Corporation. Copyright © 1992, Liant Software Corporation.

RM/PLUSDB LINKS COBOL WITH 4 GLS

RM/plusDB is Liant's bi-drectional gateway between RM/COBOL-85 applications and third-party relational database systems, it's another of the alternative filing systems open to the system designer and allows users to substitute databases for the RM/COBOL-85 indexed file system without change to the source code.

RM/plusDB brings benefits in two important ways. Firstly, it protects the user's investment in COBOL programs and applications by providing access to databases on either a file-by-file basis or "wholesale". Secondly, it provides procedural muscle to augment 4 GL systems and bring additonal power and performance to time-critical parts of an application.

An RM/plusDB dictionary identifies those files stored in the database, converts record level requests to appropriate database retrievals and updates, and manages all the data conversion. The result — applications that can utilize data from either indexed files or databases.

Initially available for Informix, Oracle, and Ingres databases in the UNIX environment, RM/plusDB represents an important part of Liant's open-systems strategy. RM/plusDB bridges the generation gap between 3GLs and 4GLs.▲

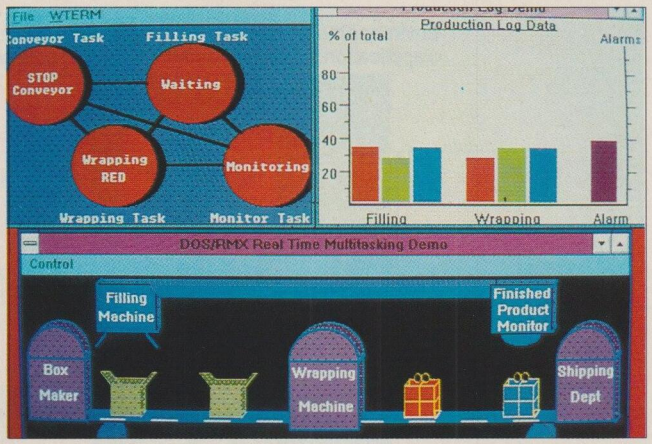
RM/COBOL REPORT-WRITER RELEASED

In September, Liant released RM/COMPANION version 1.0, the new RM/COBOL information retrieval and report generation system. Its fourth-generation features make it an ideal system for both developers and non-technical end-users. Because it was created to work specifically with RM/COBOL data files, RM/COMPANION runs faster and with lower overhead than other report writers on the market.

The powerful, English-like command language supports sorting, selection, calculation and formatting. Existing RM/COBOL datafiles can be quickly and easily searched, manipulated, and printed or displayed according to the user's specifications. The data gathered by RM/COMPANION can later be reformatted, graphed and plotted.

RM/COMPANION provides the option of generating reports in either compiled or interpreted mode for maximum flexibility. By selecting the "compiled" option, developers can generate an RM/COBOL-85 object program that can then be distributed to end-users without the RM/COMPANION runtime component. If more end-user flexibility is desired, interpreted reports can be distributed with the end-user version, allowing for report customization and new reports from the application's data dictionary.

RM/COMPANION is available on all platforms running version 5.0 or higher of RM/COBOL-85.▲



Gestion de processus, supervision de chaînes de production. DOS/RMX offre de nombreuses possibilités aux utilisateurs.

est chargé en mémoire basse. Ensuite, un programme TSR (Terminate and Stay Resident) charge sous DOS l'exécutif multitâche temps réel, iRMX III, dans l'espace mémoire inférieur au champ d'adressage d'un mégaoctet. Dès qu'iRMX se trouve chargé, il encapsule le DOS.

DOS et iRMX, un pont permanent

Ce dernier fonctionne alors en tant que tâche d'iRMX (tâche VM86) disposant du niveau de priorité le plus bas. Il devient alors possible de charger MS Windows et de passer d'un écran iRMX à un écran Windows, par différentes combinaisons de touches. L'utilisateur peut, par exemple, se servir iRMX pour écrire en temps réel des données dans un écran de Windows.

Il dispose ainsi, sur son écran, de données graphiques ou de tableaux de chiffres directement mis à jour à partir des informations saisies en temps réel. DOS et iRMX peuvent donc échanger des données par l'intermédiaire de fichiers et synchroniser leurs activités.

La version actuelle de DOS/RMX ne permet pas de faire tourner les logiciels tels que Lotus 1-2-3 ou encore Excel 3.0 qui utilisent le processeur de l'ordinateur en mode protégé.

Aucune application utilisant un gestionnaire de mémoire haute ne fonctionne actuellement sous DOS/RMX. Pour y remédier, une nouvelle version de DOS/RMX a été développée en collaboration avec Microsoft permettant ainsi le développement avec les versions bêta de Windows 3.1.

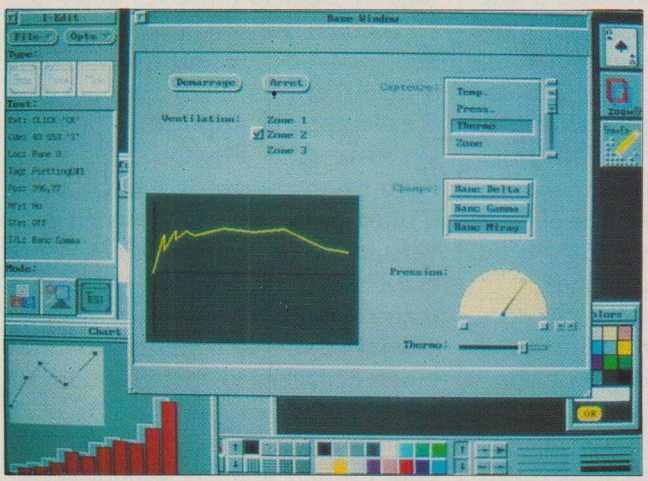
Cette Version 2.0, renommée «Windows/RMX», devrait sortir dans les prochains mois et être entièrement compatible avec le DOS 5.0 et Windows 3.1.

iRMX fournit un nombre important de fonctions temps réel, accessibles à la programmation telles que la gestion des applications multitâches, la gestion préemptive des priorités ou encore la communication entre les différentes tâches à l'aide de boîtes aux lettres et de sémaphores. DOS/RMX est compatible avec les réseaux MS-NET, iRMX-NET et supporte également Novell Networking et Microsoft LAN Manager. Il offre aussi un environnement de développement qui comprend des langages de programmation (C, PL/M et assembleur) et un programme hypertexte DOS d'aide au développeur. Le package comprenant la licence DOS/RMX, le compilateur C 386, les outils 386 et un éditeur de textes est disponible pour 17 122 F HT. Le système DOS/RMX, seul, vaut quant à lui : 8 856 F HT. A noter que le montant des royalties à reverser à Intel en cas de commercialisation d'une application utilisant DOS/RMX se révèle relativement modeste : 886 F HT.

FlexOS, accéder à l'espace mémoire

Depuis 1985, Digital Research propose un système d'exploitation pour PC, multitâche, multi-utilisateur et temps réel. Son nom : FlexOS, pour Flexible Operating System. FlexOS autorise le fonctionnement du processeur i186 en mode réel et des i80286 et i80386 en mode protégé.

Il offre donc l'avantage de pouvoir accéder linéairement à une RAM plus importante. FlexOS 286, pour processeur i286, permet d'adresser jusqu'à 16 Mo d'espace mémoire. FlexOS 386 offre, quant à

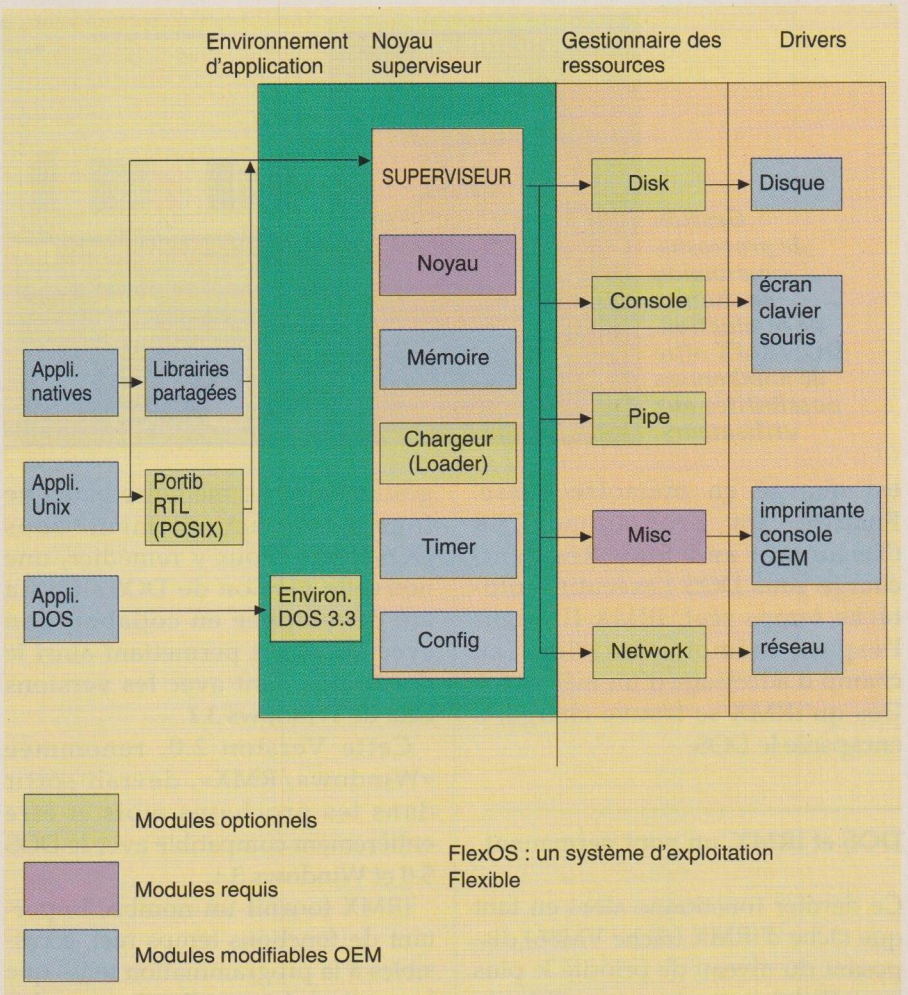


Interface Editor : un éditeur révolutionnaire sous QNX. Boîte à outils, menus déroulants ... Il permet à l'utilisateur de créer des interfaces à ses programmes sans écrire la moindre ligne de code.

lui, la possibilité d'adresser un maximum de 2 Go. Tous deux permettent d'assurer la gestion et la protection de la mémoire sous le contrôle du système d'exploitation. FlexOS contient un noyau de petite taille, nommé «superviseur», totalement implantable en mémoire morte.

Ce superviseur gère l'ensemble des communications entre les applications et les gestionnaires des ressources du système. Ainsi, aucune application n'a accès directement au hardware, mais chacune utilise les 34 appels système mis à sa disposition par le superviseur.

Les gestionnaires de ressources sont, en fait, des modules indépendants qui communiquent avec le superviseur. Disk, par exemple, s'occupe de la gestion des unités disques (gestion des fats, des clusters...), Console administre les consoles virtuelles, un autre, nommé «Drivers», permet d'attacher un driver quel qu'il soit, série ou parallèle.



Partager la mémoire

Il est à noter que les drivers peuvent être chargés dynamiquement par un applicatif ou simplement à partir d'une ligne de commande. Le Loader du superviseur autorise lui, l'utilisation de librairies partagées.

FlexOS permet la création d'une zone mémoire partagée pouvant être mise à jour ou consultée simultanément par plusieurs applications. Il offre la possibilité d'attribuer des niveaux de priorité aux différentes tâches et de les faire communiquer, en envoyant des messages par l'intermédiaire de «pipes» ou canaux d'intercommunication.

Par ailleurs, FlexOS n'est pas sans rappeler DOS et Unix puisqu'il reprend un certain nombre de commandes DOS (DIR, CHKDSK, CD, DEL, FORMAT...) et offre également des utilitaires du monde Unix (CAT, Banner, Diff, Dump, FGREP...). Chaque PC peut s'initialiser à la fois sous FlexOS et sous DOS. FlexOS supporte les disques et les systèmes de fichiers du DOS et autorise le partage de fichiers entre les deux OS. FlexOS 386, version pour processeur 386, possède un environnement d'applications DOS qui permet d'exécuter des applicatifs pour ce système sous FlexOS. FlexNET, l'extension réseau de FlexOS offre d'intégrer des informations en provenance de différents systèmes Unix et DOS notamment par l'intermédiaire du support NetBios.

Sous FlexOS, le temps de commutation de tâches (dispatch) atteint 60 microsecondes sur un 386/25 (contre 145 microsecondes à QNX pour un 386/16 microsecondes et 50 pour un 386/33). Le temps de latence moyen approche les 18 microsecondes.

FlexOS propose également un environnement de développement complet (compilateurs, éditeurs de liens, débogueurs...). Il possède, en plus, une interface graphique X/GEM qui, si elle n'affecte en rien les capacités temps réel de FlexOS et facilite le développement des applications, n'en demeure pas moins un peu triste.

SIM/2, MS-DOS amélioré

SIM/2 est édité par Software et Systèmes Spéciaux (France), et distribué par Microprocess sous le nom de S-DOS. Nul besoin d'ému-

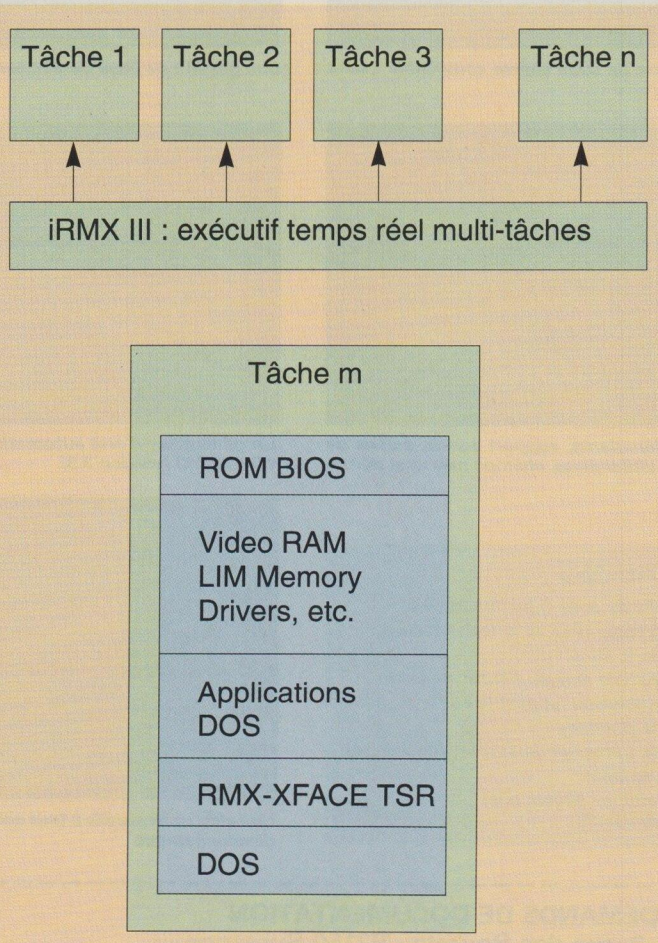
lateur DOS pour S-DOS : le système d'exploitation multitâche, temps réel, choisi par IBM est 100% compatible avec MS-DOS. Pour créer un tel système, la société française 3S a réécrit, et amélioré, les principales fonctions de MS-DOS. Elle y a intégré plus de 200 primitives liées au temps réel et au multitâche (création de processus, gestion des timers, réservation des ressources...).

Ainsi, les fonctions de MS-DOS telles que la lecture ou l'écriture sur le disque ne monopolisent plus la machine. De telles opérations sont dorénavant «partageables». SIM/2, nom choisi par IBM pour S-DOS, peut être installé sur tous les ordinateurs de type PC, du 8088 au 486, en passant par le PS/2.

Il est compatible avec les applications DOS standard : les fonctions système, les logiciels du commerce (Multiplan, Word...) ou encore les compilateurs et les débogueurs de Microsoft et Borland, incluant des librairies standards. Les fonctions de gestion de fichiers de type MS-DOS sont toutes présentes sous SIM/2.

Trois types de fichiers sont exécutables : les .EXE et les .COM de MS-DOS, ainsi que les .EXE de S-DOS. Les commandes internes (IF, GOTO...) et externes (COPY, TIME...) permettent de créer des fichiers .BAT compatibles avec ceux de MS-DOS.

La taille de SIM/2, d'environ 64 ko, lui confère un avantage certain : pouvoir être entièrement stocké en mémoire morte. Il laisse ainsi la mémoire libre aux applications et retarde d'autant le recours à des cartes d'extension mémoire.



DOS encapsulé par iRMX

DOS/RMX : la réunion du DOS de Microsoft et d'iRMX d'Intel

Briser la barrière des 640 ko

Autre avantage, par rapport au MS-DOS classique, SIM/2 passe outre la barrière fatidique des 640 ko et peut gérer jusqu'à 16 Mo de mémoire vive. SIM/2 utilise les cartes d'extension mémoire XMA d'IBM et AMB d'Olivetti. De plus, la taille des fichiers et des unités de disque n'est plus limitée. Pour les fichiers, il apporte une protection individuelle supplémentaire inspirée d'Unix (propriétaire, membre du groupe, autres).

Multitâche, SIM/2 permet de faire exécuter simultanément plusieurs programmes écrits pour MS-DOS ou pour SIM/2 lui-même. Le nombre de processus créés dynamiquement se trouve uniquement limité par la taille mémoire. SIM/2 distingue 3 types de tâches, selon leur niveau de priorité :

- ❏ les traitements temps réel critiques,
- ❏ les processus normaux traités en temps partagé,
- ❏ les tâches de fond.

Les tâches de même niveau s'exécutent tour à tour sur la base d'une priorité cyclique.

La synchronisation et la communication entre processus s'effectuent par envoi de messages, boîtes aux lettres et sémaphores. SIM/2 fait une utilisation accrue des mémoires partagées qui permettent le partage de code entre plusieurs processus tournant en parallèle. Le gestionnaire de mémoire crée et libère dynamiquement des parti-

DATABOSS: LE GENERATEUR COMPLET DE VOS APPLICATIONS EN PASCAL OU C

Avec DATABOSS, gagnez des semaines de développement.

DATABOSS est un environnement complet conçu pour le développement de bases de données relationnelles pour des applications en Turbo Pascal ou en C.
DATABOSS permet de générer les différentes parties de votre application: menus, fichiers (avec linkage), écrans d'entrées de données, état, réindexation et modules de reconfigurations, rapports.

DATABOSS produit du code source Pascal ou C pour des applications mono ou multi-utilisateurs. Les écrans peuvent être multi-fenêtres, comporter des champs mémos, ainsi qu'une aide en ligne et des champs virtuels ou calculés.

Son nouveau générateur d'état WYSYWYG vous permet de créer vos rapports très facilement.

De plus DATABOSS ne nécessite aucune connaissance de développement.
Sa documentation en français comporte plus de 300 pages et le support technique est gratuit.

La version de démonstration est limitée à 10 utilisations et l'ensemble des manuels est disponible à **490 F HT.**

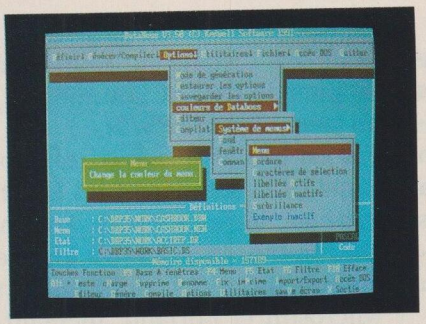
Version 3.0 en C. et 3.5 en Pascal

Prix intro: ~~7 995~~ F HT
5 995 F HT TVA 18,6%
jusqu'au 31/05/92

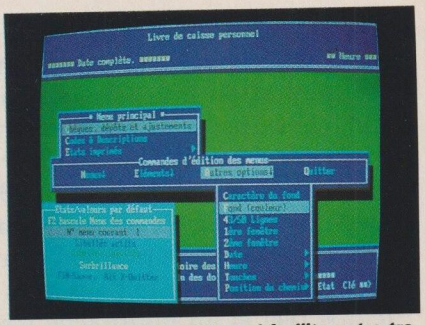
Pour vous renseigner ou commander
Téléphonez au: (1) 40 99 28 00
Fax: (1) 40 99 28 88

INNOSOFT

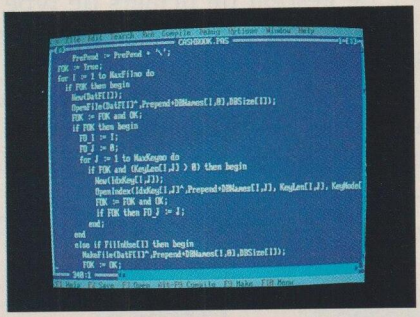
2, rue des Bourets - 92150 Suresnes



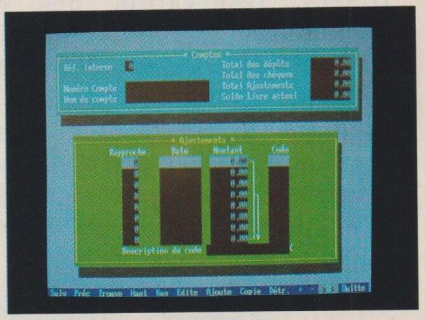
Un générateur complet.



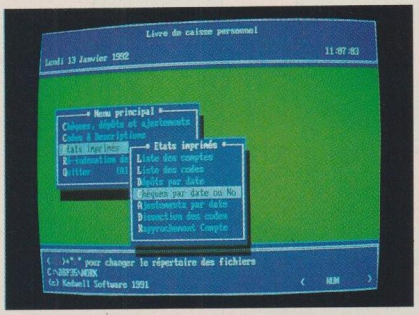
Une interface conviviale qui facilite votre travail.



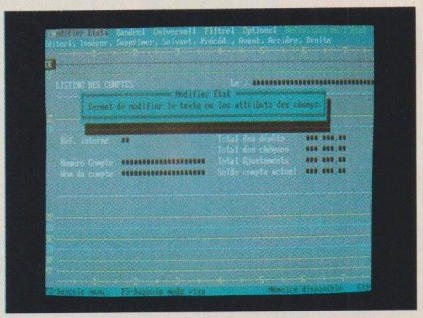
Génération de code source compilable - sans royalties.



Une gestion de base de données ISAM.



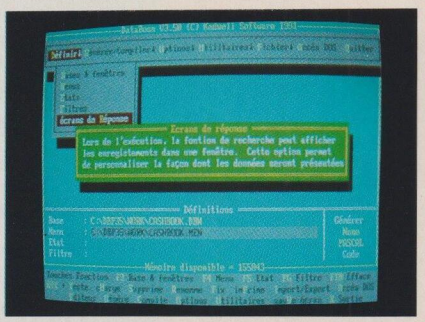
Menus déroulants, support souris, écrans de saisie, multifenêtres, champs mémo et déroulables.



Un générateur d'état automatique en WYSYWYG (version 3.5).

DATABOSS

- *Génère du code C ou Pascal*
- *Compatible TP55 et 6, Quick Pascal, Turbo C, C++*
- *Compatible Novell, 3 COM, Netbios*
- *Générateur de rapport WYSYWYG (3.5)*
- *Pas de royalties*
- *Pas de nécessité de savoir programmer*
- *Aide en ligne*
- *Importation directe possible depuis dBASE.*



Une aide contextuelle à tout niveau simplifie le développement.

Service lecteurs n° 20 (dernière page)

BON DE COMMANDE/DEMANDE DE DOCUMENTATION
A renvoyer à INNOSOFT - 2, rue des Bourets - 92150 Suresnes

☐ je commande **la version d'évaluation de DATABOSS** avec ses manuels au prix de **652,30 F TTC** (soit 581,14 F TTC + 71,16 F TTC de frais de port).
☐ **OUI** je désire recevoir une documentation commerciale détaillée.

Ci-joint mon règlement:
☐ Chèque ☐ Carte Bleue N° ..... Date d'expiration.....
☐ Contre-remboursement (60 F TTC en sus).

Nom: ..... Adresse: .....
Société: ..... Code postal: .....
Ville: ..... Tél.: .....

LS7

tions et gère l'extension mémoire LIM EMS du DOS. SIM/2 supporte également les fonctions de réseau et d'appel à distance.

Pour cela, il utilise un gestionnaire de réseau local s'appuyant sur Netbios et employant le protocole SMB. L'utilisateur dispose alors d'un système d'exploitation distribué de type client-serveur entre plusieurs PC.

Question performances, les temps de réponses avec changement de contexte (temps de commutation de tâches) sont de l'ordre de 150 microsecondes sur un 386/16 (contre 145 microsecondes pour QNX). Le temps de latence moyen (durée de la réponse à une interruption) atteint, quant à lui une dizaine de microsecondes.

Le kit de développement SIM/2 comprend : une licence de développement, 5 jours d'assistance-formation, une documentation et une version run time. SIM/2 est soumis à un tarif dégressif, allant de 40 000 F HT pour un à quatre utilisateurs à 20 000 F HT pour plus de cent usagers.

La version run time seule est disponible à partir de 1 500 F HT pour un à neuf utilisateurs. Tous ces résultats ont incité IBM à insérer S-DOS à son catalogue, sous le nom de SIM/2 (Système industriel multitâche de niveau 2). Il viendra prendre place dans l'offre CIM (Computer Integrated Manufacturing) d'IBM au niveau 2 (supervision, pilotage d'atelier ou contrôle de qualité) et complétera le logiciel PFSACE développé pour le niveau 1 (automate et contrôleur de robots).

QUELQUES DÉFINITIONS

☛ **Multiposte :** permet de connecter plusieurs couples écrans- claviers (postes) qui travailleront simultanément sur une même unité centrale.

☛ **Multitâche :** plusieurs programmes peuvent s'exécuter simultanément à partir d'un même poste. On distingue généralement le multitâche vrai du multitâche classique. Dans les cas du multitâche vrai, l'ordinateur possède plusieurs processeurs.
Chacun d'entre eux se trouve affecté à une tâche particulière, et un processeur maître s'occupe de la gestion générale de l'ensemble. Pour le multitâche classique, en revanche, un unique processeur gère plusieurs tâches et partage son temps entre elles selon leur niveau de priorité. Il n'accomplit en fait qu'une petite partie de tâche à la fois en un temps très court. Il donne ainsi l'impression de simultanéité des différents processus exécutés.

☛ **Multi-utilisateur :** permet l'accès à un même fichier, au même instant, à plusieurs utilisateurs, tout en respectant l'intégrité et la confidentialité des données.

☛ **Réparti :** toutes les fonctionnalités du système d'exploitation peuvent être activées depuis une même machine, pour être exécutées sur une autre, ceci de façon transparente pour l'utilisateur.

☛ **Temps réel :** capacité de réagir quasi instantanément à un événement. Un système temps réel doit pouvoir traiter les informations au fur et à mesure qu'elles se présentent, qu'elles soient espacées d'une seconde ou d'une dizaine de microsecondes. Il se traduit par des temps de commutation très rapides d'un process à un autre et par un temps de réactivité très court aux événements extérieurs.
Pour mesurer la vitesse de commutation des tâches, on utilise généralement deux tâches qui jouent au ping-pong avec un octet. On mesure alors le temps de commutation d'une tâche à une autre.
Ceci inclut l'envoi, le passage par le dispatcher et la réception. La réactivité à un événement extérieur se mesure d'habitude en calculant le temps de latence moyen du système (average latency time).
Pour cela, on mesure le temps écoulé entre l'envoi d'une interruption (qui traduit l'arrivée d'un événement extérieur) et le début réel de la tâche correspondant à cet événement. Pour des systèmes temps réel, ces durées ne doivent pas dépasser quelques dizaines de microsecondes.

Realtime Craft

Realtime Craft, édité par GSI-Tecsi (France), est la première gamme européenne de systèmes d'exploitation temps réel multitâche pour PC. Disponible depuis 1985 en deux versions, 8086 et 80386 en mode protégé, elle est entièrement conforme à la norme Sceptre (Standardisation du cœur des exécutifs des produits temps réel européens) proposée en 1982 par le Boni (Bureau d'Orientation de la Normalisation de l'Informatique). Realtime Craft PC intègre tous les avantages de MS-DOS dans une solution temps réel multitâche.

Realtime Craft permet d'exécuter en parallèle des applications MS-DOS et des tâches temps réel qui répondent à des événements extérieurs : informations capteurs, interruptions...

Realtime Craft 86, tout d'abord, constitue un système conçu pour les microprocesseurs Intel 8086. Il comporte un exécutif multitâche XEC 86, un traceur TRC 86, sorte de débogueur spécifique (qui autorise la mise au point des routines multitâches), un gestionnaire d'écrans multifenêtres, multitâches Screen Manager, un gestionnaire d'applications curieusement nommé «Exorsyst» et divers

utilitaires supplémentaires dont le source est fourni.

Les produits RTC se présentent sous la forme de modules indépendants. L'utilisateur accède aux services de ces «composants logiciels» en invoquant un point d'entrée prédéfini, ou encore par interruption logicielle.

Une interface conviviale

Un chargeur-installateur, équivalent du Loader sous d'autres OS, permet de les sélectionner, de les configurer et de les charger en tant que programmes résidents, le tout à travers une interface conviviale. XEC 86, l'exécutif temps réel situé au cœur de RTC 86, gère l'attribution des ressources du microprocesseur entre les différentes tâches. XEC est très compact (moins de 3 ko).

Il peut donc être entièrement implanté en mémoire morte. Il offre au développeur 27 commandes qui autorise l'accomplissement de l'ensemble des opérations systèmes nécessaires aux applications temps réel.

Parmi ces différentes opérations, on retiendra tout d'abord la gestion multitâche qui correspond à la gestion des tâches, de leur état et de leur niveau de priorité ensuite, la protection et le partage des ressources, la synchronisation et la communication entre tâches (sémaphore, boîte aux lettres) et, enfin, la gestion du temps, des événements et des interruptions.

Le temps de changement de contexte sous RTC est de 62 microsecondes sur un 8086 cadencé à 8 MHz et de 31 microsecondes sur un 80386/16. Le temps de latence moyen varie quant à lui en fonction du «service» requis et de l'interruption logicielle correspondante.

La solution Realtime Craft fait cohabiter l'exécutif XEC et MS-DOS. XEC est chargé en tant que code résident dans la RAM du PC. MS-DOS, pour sa part, reste totalement passif, et non réentrant.

Une application peut donc, sous le contrôle d'un utilitaire de protection, continuer d'appeler tous les services de MS-DOS, du Bios et des gestionnaires de périphériques déjà disponibles sur son ordinateur. Simultanément, elle a également la possibilité de faire appel à tous les services offerts par le noyau XEC qui lui permettront de gérer et de créer des tâches.
L'utilisateur peut, par exemple, accéder à ses fichiers en invoquant et en protégeant les services standards de MS-DOS.

Il rend ainsi le gestionnaire de fichiers de MS-DOS multitâche. A noter que FMS/DOS, le gestionnaire de fichiers au format MS-DOS, est un module complémentaire qui gère des accès directs et séquentiels à des fichiers dans un environnement multitâche et en temps réel. La gestion hiérarchique des répertoires y est assurée.

Par ailleurs, une application développée sur MS-DOS sera récupérée sans modification de code, dans l'environnement Realtime Craft 86 sous... MS-DOS. Egalement, le gestionnaire Exorsyst effectue le chargement en mémoire des différentes applications ainsi que leur lancement en parallèle.

Pour les tâches qui nécessitent plus de 640 ko, Exorsyst gère les cartes d'extension mémoire au format IBM XMA, permettant ainsi d'atteindre 4 Mo.

Exploiter les avantages du 386

Realtime Craft 386 permet de tirer parti des avantages du 386 en mode protégé : notion de tâche intégrée dans le microprocesseur, protection de la mémoire et adressage étendu jusqu'à 4 Go. Il comporte notamment un exécutif temps réel, un traceur, un moniteur de machines virtuelles et des gestionnaires de mémoire, d'objets et d'entrées-sorties.

Si les procédures de configuration et de chargement sont identiques à celles de RTC 86, il faut signaler que le format Intel, pour les exécutables 386 (OMF 386), ne peut pas être chargé par MS-DOS.

Pour cette raison, GSI-Tecsi a développé un utilitaire de chargement d'exécutable au format OMF 386 qui se trouve débarrassé des restrictions d'adressage propres à MS-DOS.

Cet utilitaire peut être appelé directement à partir de l'interpréteur de commandes MS-DOS. En ce qui concerne l'exécution, le moniteur de machine virtuelle, VMM 386, permet à une application de faire exécuter en parallèle plusieurs tâches 386 en mode pro-

ATTENTION ! MULTITACHE MULTI-UTILISATEUR NE SIGNIFIENT PAS TEMPS RÉEL!

Multitâche multi-utilisateur ne signifient pas pour autant temps réel! Il suffit de travailler sous Unix pour s'en convaincre... Il ne faut pas non plus confondre système temps réel et système transactionnel! Ils sont tous deux généralement multipostes, multitâches, multi-utilisateurs...

Mais les systèmes transactionnels ont des temps de commutation de tâches et des temps de latence moyens nettement supérieurs à ceux des systèmes temps réel. De plus, ils occupent habituellement une place en mémoire beaucoup plus importante. Les uns seront donc orientés gestion, les autres vers des applicatifs plus «industriels».

tégé (32 bits), et d'autres, comme l'interprêteur de commande MS-DOS, en tant que tâche virtuelle 8086 (16 bits).

Realtime Craft est disponible au prix de 50 000 F HT pour 6 copies de développement.

La version run time fait, elle, l'objet d'un tarif dégressif selon le nombre de copies.

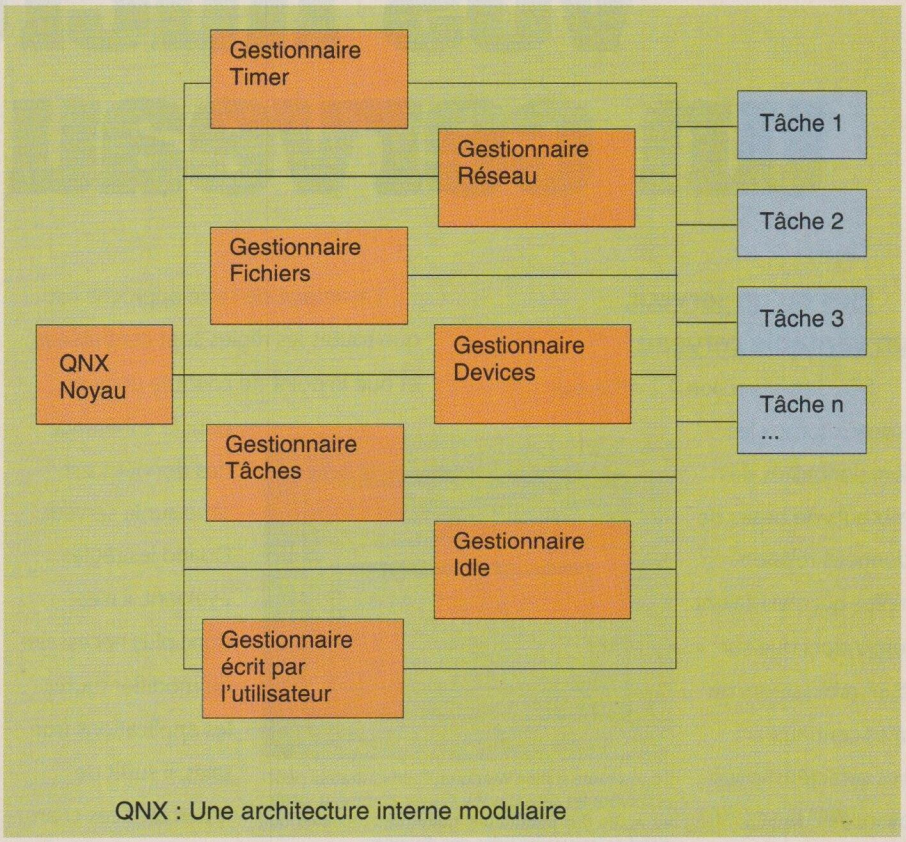
Chorus
l'incontournable

Comment parler de systèmes d'exploitation sans citer Chorus? Son OS, développé par Chorus Systems (France), a été choisi par AT&T pour être le standard du temps réel sous Unix! Tout comme QNX, Chorus constitue un système d'exploitation temps réel construit autour d'un micronoyau.

Celui-ci gère un ensemble d'unités ou de serveurs qui réalisent les fonctions nécessaires à un système ouvert. Il est conçu pour gérer les communications à l'intérieur du système d'exploitation et pour fournir les services de base sur lesquels sont bâtis les serveurs indépendants.

Il comporte un exécutif temps réel, des mécanismes de communication interprocess, un superviseur qui contrôle les ressources matérielles et un dispositif de gestion de mémoire virtuelle répartie. L'exécutif temps réel assure l'ordonnancement préemptif des tâches selon leur niveau de priorité. Il existe, sous Chorus, 256 niveaux de priorité accessibles à la programmation.

De plus, ce système d'exploitation est entièrement modulaire et reconfigurable dynamiquement, puisque les nouvelles fonctions se trouvent contenues dans un serveur indépendant. Chorus offre un sous-système Unix constitué d'un ensemble de serveurs supportés par le noyau. Ce sous-système offre



QNX : Une architecture interne modulaire

à l'utilisateur, et au développeur, un environnement Unix modulaire, temps réel, réparti et conforme aux standards internationaux (System V - X/Open).

Il réalise chacune des fonctions d'un noyau Unix traditionnel : gestion des processus, des fichiers, des périphériques, des tubes et des sockets. Les performances de Chorus sur PC 386/25 sont plus qu'honorables.

Le changement de contexte s'effectue en 44 microsecondes en mode système (les tâches appartiennent au même espace mémoire) et en 68 microsecondes en mode utilisateur (les tâches sont activées dans des espaces mémoire différents). Le temps de latence atteint, lui, une moyenne de 20 microsecondes.

S'il tourne bel et bien sur PC, Chorus n'est, à ce jour, toujours pas commercialisé.

Pourquoi tenir un tel système d'exploitation enfermé dans une machine dite de référence et d'évaluation? *«Question de stratégie marketing et de Packaging»* nous répond-on chez Chorus Systems.

Il est vrai que la version PC actuelle de Chorus ne dispose d'aucune compatibilité avec MS-DOS et de peu (voire pas du tout) d'utilitaires et de drivers disponibles.

De plus, la firme Chorus Systems ne vise pas la commercialisation immédiate de ses binaires, mais plutôt la fourniture à des intégrateurs-développeurs des sources d'un système temps réel compatible Unix pour des machines parallèles hautes performances et supportant la tolérance de pannes.

Soit, mais pendant ce temps, les OS précédemment cités envahissent le monde de la micro.

Et bon nombre d'entre eux s'enorgueillissent déjà de plusieurs centaines de milliers de postes installés sous leurs systèmes d'exploitation respectifs. ❑

Nouveau Microsoft vos micros vont un cerveau particu

Plus qu'un serveur, un véritable cerveau !

SQL Server version 4.2 offre aux réseaux locaux les fonctionnalités des moteurs de bases de données relationnelles qui n'existaient jusqu'alors que sur mini-ordinateurs ou sites centraux.

Les caractéristiques avancées de SQL Server 4.2 en font, dans l'univers de la micro-informatique, le seul serveur de bases de données relationnelles intelligent.

Depuis toute station Windows, l'administrateur peut superviser les serveurs SQL Server du réseau, gérer les accès des utilisateurs, traiter des requêtes ou planifier les sauvegardes des données.

L'intelligence de SQL Server permet d'incorporer au sein même de la base de données, les règles d'intégrité des données et les contraintes propres à l'entreprise, plutôt que d'avoir à programmer chaque application frontale pour qu'elle respecte ces différentes règles.

L'avantage de cette approche est que toutes les règles sont centralisées et que le point de contrôle unique pour la cohérence des données est situé sur le serveur. Quand les règles évoluent, il n'est donc plus nécessaire de modifier toutes les applications frontales, il suffit de répercuter ces changements sur le serveur. Le développement et la maintenance des applications s'en trouvent simplifiés et la sécurité des données sera effectivement assurée quelle que soit l'application frontale utilisée.

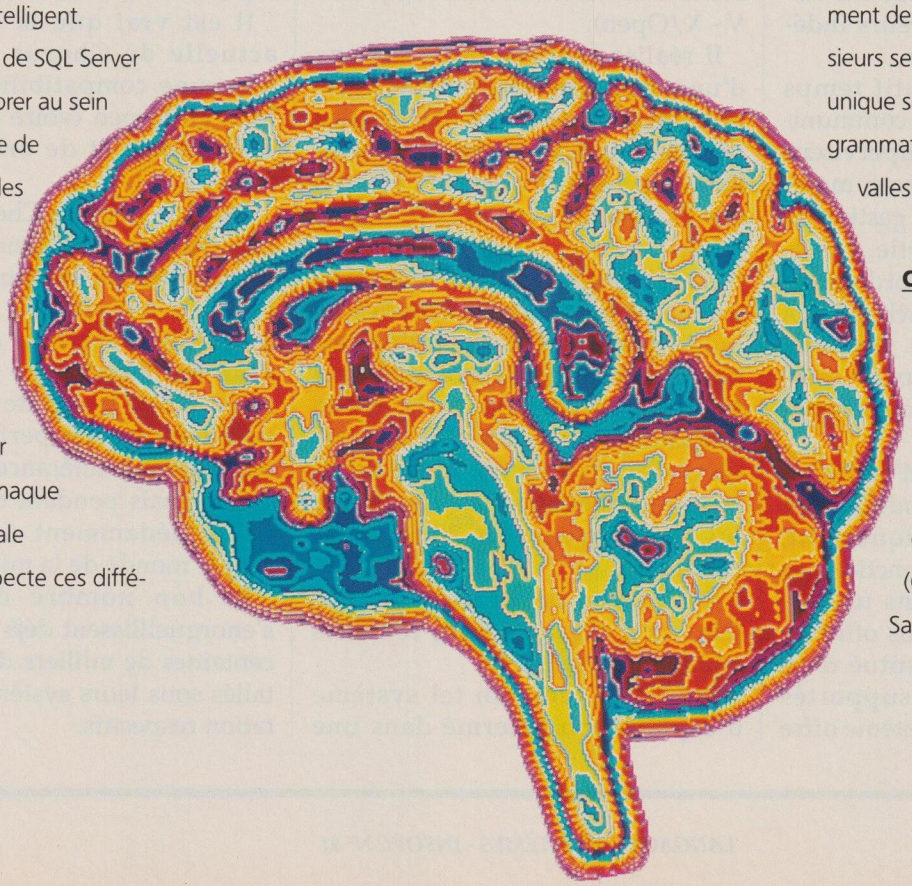
SQL Server 4.2 est conçu pour opérer 24 heures sur 24 dans des applications critiques, et permet de faire des sauvegardes dynamiques sur bande ou sur disque alors que les utilisateurs continuent de travailler.

Un outil d'administration complet sous Windows.

SQL Server 4.2 propose un nouvel outil d'administration qui permet de gérer les utilisateurs, les permissions d'accès, la sauvegarde des données et de l'environnement. Il autorise également des connexions simultanées à plusieurs serveurs depuis une station unique sous Windows, ainsi que la programmation de sauvegardes à intervalles réguliers.

Le nec plus ultra de la connectivité.

SQL Server 4.2 permet de s'interfacer avec des bases de données relationnelles ou non relationnelles situées sur minis ou grands systèmes grâce à ses services ODS (Open Data Services). Sans qu'il soit nécessaire de



ROUX SEGUELA CAYZAC & GOUDARD Photo Cosmos

SQL Server 4.2 : être en liaison avec lièrement brillant.

modifier une application frontale accédant à SQL Server, il est possible de permettre un accès transparent à des bases de données externes ou encore de déclencher l'exécution d'un programme prédéfini.

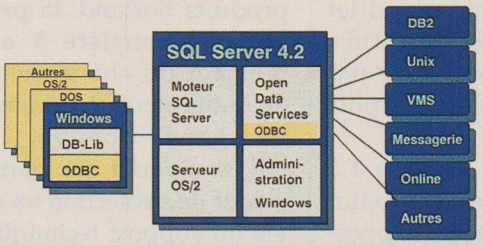
En cas de modification du niveau d'un stock dans la base de données, par exemple, SQL Server déclenchera automatiquement l'envoi d'un message à la personne responsable des achats via la messagerie électronique de l'entreprise, si le niveau du stock tombe en-dessous d'un seuil prédéterminé.

Outre Microsoft LAN Manager, SQL Server est compatible avec les plus grands systèmes d'exploitation réseaux du marché. Le protocole de communication de SQL Server a en effet été adopté par les principaux éditeurs de produits réseaux : Novell Netware, IBM LAN Server...

Une ouverture en toute sécurité.

SQL Server 4.2 est ouvert à un très grand nombre d'applications frontales, permettant ainsi aux utilisateurs de conserver l'environnement qui leur est familier.

Les coûts de formation en sont sensiblement réduits et l'investissement initial de l'entreprise n'est pas remis en cause.



A travers les services ODS de SQL Server 4.2, les stations clientes accèdent de façon transparente à des applications existantes telles que des bases de données relationnelles ou non, des services de messagerie...

A ce jour, plus de 90 applications ont été répertoriées autour de SQL Server. Parmi les applications principales on trouve notamment : Microsoft Excel, Visual Basic, SQL, Windows, dBase, Paradox, Lotus 1-2-3, Superbase, Omnis 5, Objectview...

Cette ouverture exceptionnelle est de plus complètement sécurisée étant donné que SQL Server gère lui-même l'intégrité des données plutôt que de laisser ce soin aux différentes applications frontales utilisées.

La plate-forme de production micro.

SQL Server 4.2, grâce à ses caractéristiques de sécurité, d'ouverture et de performances, représente une réelle plate-forme de production dans le domaine de la micro-informatique. Cette plate-forme présente des performances similaires à celles des architectures multi-utilisateurs classiques et ce pour un rapport prix/performance particulièrement avantageux du fait de l'utilisation de micro-ordinateurs standards du marché.

Pour toutes informations complémentaires ou recevoir la liste des Revendeurs Spécialistes Réseaux Microsoft :
- appelez le service Clients au (1) 69 86 46 46
- ou utilisez le Minitel : 3616 code Microsoft.

Microsoft, 91957 Les Ulis Cedex
et à Infomart La Défense
Stand n° 276 niveau 2.

Microsoft®

LES LOGICIELS QUI DONNENT DES AILES.

Toutes les marques citées sont des marques déposées.

LANGAGE

LES APPORTS DE dBASE IV 1.5

Une version revue par Borland

ARNAUD DE LA POMMERAYE

dBase IV 1.5 est le successeur de la version 1.1 sortie en juillet 1990. Cette nouvelle mouture — revue, corrigée et embellie par Borland — est intéressante à plus d'un titre.

En premier lieu, la version 1.5 révèle les éléments clés du produit sur lesquels il était urgent d'apporter des modifications, tâche d'autant plus difficile pour Borland que dBase est un «lourd héritage» en matière de code. Ensuite, parce que cette nouvelle variante laisse entrevoir les futurs développements que l'éditeur effectuera sur ce SGBD. Enfin, cette rapide analyse tentera de clarifier l'intérêt que représente dBase IV pour Borland dans sa stratégie de produits.

Installation simplifiée et support de la souris

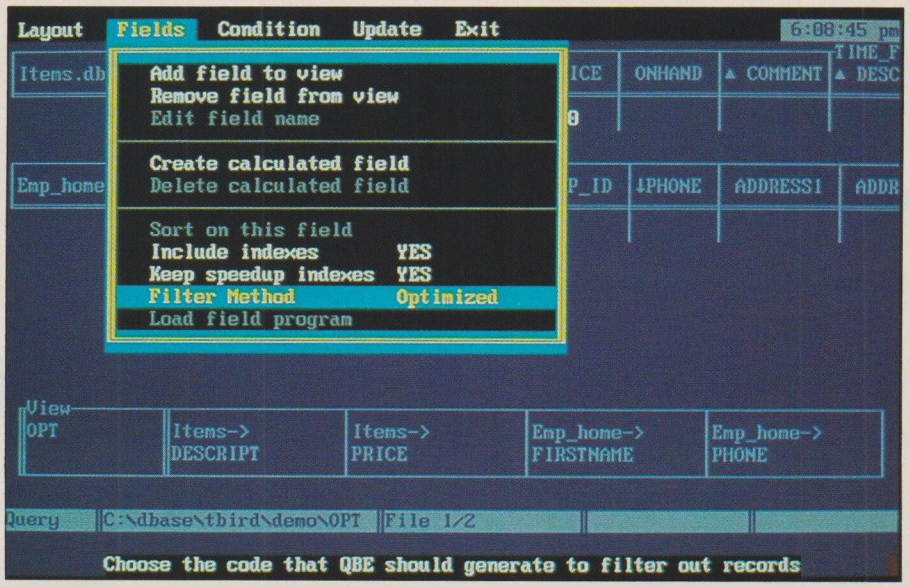
La procédure d'installation de dBase IV est désormais plus simple et plus rapide qu'avec les précédentes versions. L'utilisateur peut choisir entre trois types d'installation : rapide (le programme se charge de tout), complète (permettant le choix de divers paramètres) ou personnalisée (pour un plus grand contrôle de la procédure). L'interface utilisateur de cette procédure — qui tient compte les paramètres propres à la machine — reprend des éléments du centre de contrôle tels que les lignes d'état et de messages. Suivant en cela les produits Borland, la protection du logiciel consiste à afficher au moment du chargement l'identité de l'utilisateur et le numéro de série du logiciel.

Il est à parier que ce nouveau dispositif de protection va simplifier la vie du support technique tant il est vrai que beaucoup de disfonctionnements de la version 1.1 étaient issus de la protection. Depuis sa première version, le centre de contrôle de dBase IV constitue le cœur du système de gestion des données.

Il n'a pas connu de lifting. La seule nouveauté réside dans le support de la souris. Cette dernière permet d'effectuer la plupart des opérations de sélection dans les menus comme dans les boîtes de dialogue sans avoir à taper une touche de fonction ou un raccourci-clavier.

Elle est également très pratique en mode consultation pour modifier la largeur des colonnes ou dans le module d'extraction de données pour sélectionner des champs à inclure dans la vue. Plus importante est la conservation de l'existant. Les utilisateurs peuvent se rassurer : dBase IV 1.5 est totalement compatible avec les fichiers conçus sous dBase III Plus et dBase IV 1.0/1.1, de même que pour les applications.

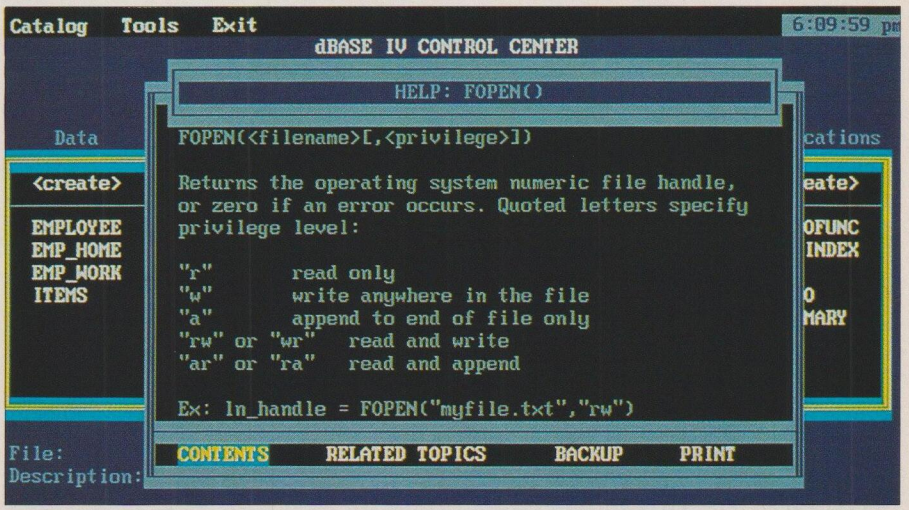
Mieux, il leur offre plus d'espace pour travailler. Dorénavant, l'utilisateur dispose de 40 «zones de tra-



dBase IV 1.5 est totalement compatible avec les versions précédentes.

La récente sortie de dBase IV version 1.5 et l'annonce de la disponibilité très prochaine d'un compilateur dBase destiné à l'environnement Windows démontrent que Borland s'intéresse de près à dBase et qu'il a l'intention de lui donner un nouveau départ dans la vie. Ces produits auront de quoi rassurer une communauté de développeurs bien acquise au langage dBase, même si certains d'entre eux complètent leurs outils de développement avec des produits comme Clipper.

vail» pour exploiter ses bases de données (avec leurs fichiers d'index associés). Cette nouvelle valeur quadruple l'espace autorisé avec la version 1.1, limite qui était très souvent atteinte et dénoncée par les utilisateurs de la version 1.1. L'algorithme employé pour la gestion interne des index a été amélioré; désormais, chacun d'eux dispose de son propre pointeur servant à sa mise à jour.



Seule amélioration dans le centre de contrôle : la gestion de la souris.

Un QBE optimisé

Le premier atout du QBE (extraction par l'exemple) de la version 1.5 s'appelle IQ!. Ce dernier optimise l'exécution des requêtes définies par l'utilisateur. Dès lors qu'une requête est passée, IQ! se charge de sélectionner la méthode la plus rapide pour afficher les données correspondantes. Cela se traduit par une vitesse de traitement accrue à l'exécution de la requête. Si le concept appartient à Paradox, il a le mérite d'être mis en application dans dBase IV. Détail important, IQ! fonctionne dans un environnement réseau. Cet optimiseur n'est pas le seul outil à augmenter les performances du QBE.

Il constitue l'une des quatre méthodes de sélection des enregistrements que possède dorénavant le QBE. La méthode SET KEY permet, de son côté, de définir un filtre sur l'index (et non plus sur les données), ce qui permet un accès bien plus rapide aux enregistrements tout en préparant le traitement pour les futures requêtes.

En effet, le programme sauvegarde temporairement les index lors de la première requête, de sorte qu'il est inutile de les recréer pour les requêtes à venir.

La seconde nouveauté du QBE réside dans la saisie de critères de sélection pour l'établissement de liens entre plusieurs bases, y compris sur des champs calculés. Enfin, une plus grande souplesse de travail est accordée au résultat d'une requête affichée dans une vue monofichier ou multifichier.

L'utilisateur peut à présent modifier les données affichées dans les vues produites par le processus d'extraction. En matière d'impression, la nouvelle mouture de dBase IV supporte des polices de caractères et des imprimantes laser. Notons également que dBase IV 1.5 offre désormais plusieurs modèles prédéfinis de rapports, ce qui accroît leur rapidité de conception et de mise au point : disposition en colonnes, présentation sous forme de formulaires, rapport spécifique pour le mailing.

Un environnement développement amélioré

En dépit de son nouveau positionnement «utilisateur final», dBase IV 1.5 reste un puissant outil de développement avec son générateur d'applications et son langage de programmation.

LES APPORTS DE dBASE IV

Celui-ci connaît quelques améliorations et plusieurs ajouts : plus de quarante nouvelles commandes et options. Le développeur dispose dorénavant d'une quinzaine de fonctions de «bas niveau» permettant — aux programmes écrits en langage C — de manipuler toutes sortes de fichiers : FDATE(), FSIZE(), FLDCOUNT(), FCREATE(), FCLOSE(), etc. Point de mire des développeurs exigeants, la mémoire manque toujours.

C'est pour leur en donner davantage que la fonction RUN() a été modifiée. Elle libère la quasi-totalité de l'espace mémoire pour l'attribuer à l'application gourmande; seuls 10 ko sont conservés pour le noyau élémentaire du SGBD. Tout le reste est déchargé sur disque. Cette opération connue sous le nom de «swapping» est d'autant plus efficace que le disque dur est rapide. Borland veut donner au langage dBase un caractère «orienté objet». Chaque fois que l'utilisateur crée un objet comme un formulaire ou un rapport, dBase IV 1.5 génère automatiquement le code associé. Il est ainsi possible de le récupérer dans un programme pour le compléter ou le modifier. Les développeurs peuvent paramétrer le centre de contrôle en fonction de leurs besoins.

TABLEAU DES NOUVELLES FONCTIONS DE dBASE IV 1.5

• Fonctions générales

DGEN()	Choisit un modèle de génération d'applications.
CATALOG()	Indique le nom du catalogue courant.
ID()	Donne le nom de l'utilisateur (environnement réseau).
ISBLANK()	Indique si un champ, un mémo, une variable, une expression ou une cellule est à «blanc».
SELECT()	Retourne le numéro de l'espace de travail de la base.
RUN()	Ne conserve que 10 ko du code de dBase pour allouer plus de place mémoire à l'exécution d'applications.

• Fonctions gérant les nouveaux index

DESCENDING()	Indique si un pointeur d'index a été créé avec ou sans le mot-clé «Descending» .
FOR()	Indique la condition utilisée pour la création de l'index.
TAGCOUNT()	Affiche le nombre d'index actifs dans un espace de travail défini ou dans un fichier d'index multiples.
TAGNO()	Indique le numéro de l'index défini par son nom.
UNIQUE()	Précise si l'index est unique ou non.

• Fonctions bas niveau

FTIME()	Heure de dernière mise à jour.
FDATE()	Date de dernière mise à jour.
FSIZE()	Nombre d'octets dans la base.
FDLCOUNT()	Nombre de champs de la base.
FCLOSE()	Ferme une base.
FCREATE()	Crée une nouvelle base.
FEOF()	Indique la «fin du fichier».
FERROR()	Retourne la valeur d'entrée-sortie.
FFLUSH()	Vide le contenu du buffer sur disque.
FGETS()	Lit une chaîne de caractères.
FOPEN()	Ouvre une base de données.
FPUTS()	Ecrit une chaîne de caractères.
FREAD()	Lit des caractères dans la base.
FSEEK()	Recherche une valeur dans la base.
FWRITE()	Ecrit des informations dans la base.

• Fonctions SET()

BORDER	Définit le type d'encadrement.
DATE	Définit le format de la date.
DEVICE	Indique le type du périphérique de sortie sélectionné.
INDEX	Indique le nom du fichier d'index.
MDX LIBRARY	Retourne le nom de la librairie de fonctions utilisée.
MARK	Affiche le séparateur de date.
ORDER	Indique l'index maître.
POINT	Affiche le séparateur décimal.
SEPARATOR	Affiche le séparateur de milliers.
WIN of MEMO	Nom de la fenêtre utilisée pour le champ mémo.

Une meilleure ouverture

De plus, leurs programmes peuvent être exécutés avant, après ou pendant l'utilisation d'un module de conception (édition de formes, de rapports ou d'interrogations). Cette «ouverture» du centre de contrôle pour la mise au point rapide des applications est indéniablement l'un des atouts principaux de cette nouvelle version.

Comme pour un tableur connu, il est aujourd'hui possible d'associer des add-on conçus pour dBase tels qu'un générateur de rapports ou un éditeur de textes.

Il en va de même pour toute application DOS facilement appelable à partir du SGBD. Seul regret, le langage d'interrogation SQL utilisé dans dBase IV en interactif comme en programmation ne connaît aucune amélioration.

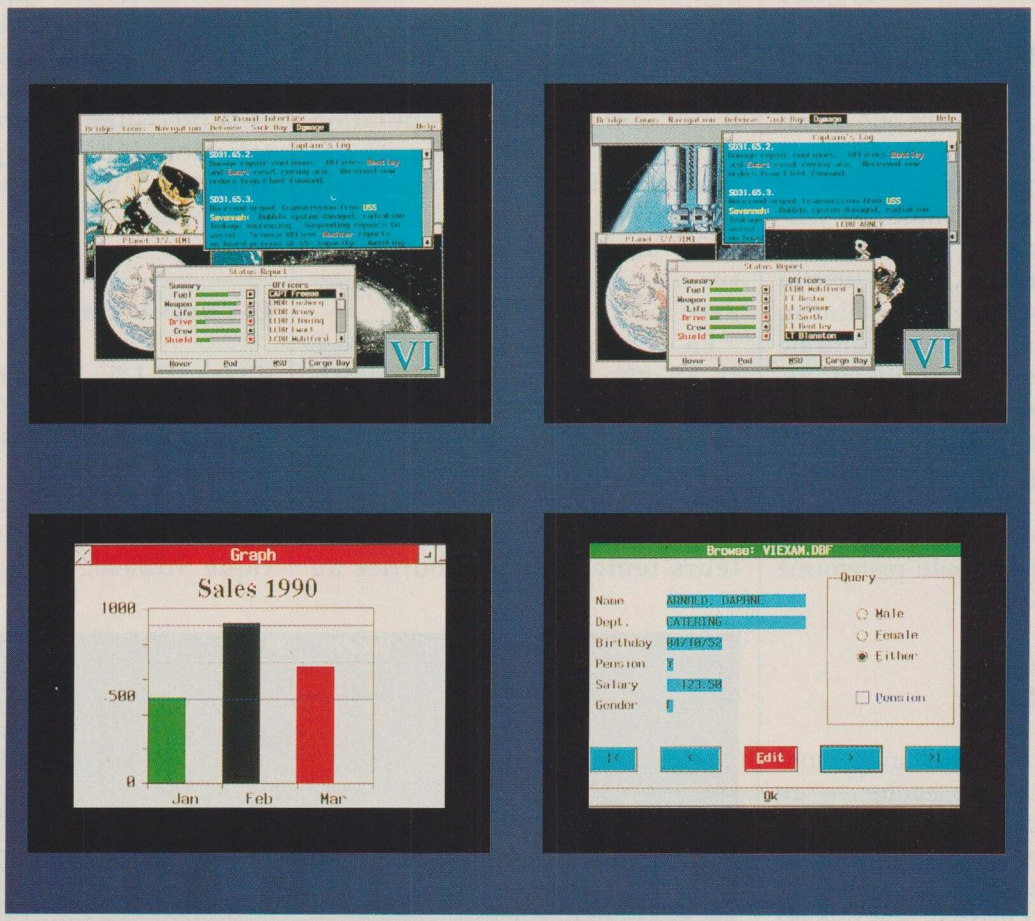
Dans les versions précédentes, on distinguait deux produits : l'un

Visual Interface pour Nantucket 5

VI (Visual Interface) pour Nantucket 5 est une interface utilisateur graphique à la Windows qui vous offre pour la première fois les 3 fonctionnalités indispensables pour vos applications :

- Des fenêtres haute résolution avec les contrôles interactifs
- Un gestionnaire d'évènement intégré
- Des fenêtres, des objets graphiques, et des icônes qui s'intègrent parfaitement à Clipper 5

Les applicatifs peuvent être exécutés à partir du DOS 3 ou plus, ou lancés à partir de Windows.



Nantucket 5 et VI : les environnements de développement en mode graphique

Je désire recevoir la documentation concernant le produit Visual Interface pour Clipper

M, Mme, Mlle .....Prénom .....

Société .....Fonction .....Adresse.....

.....

Code Postal.....Ville.....Téléphone.....

PC Technologie - 4/4 bis, rue Bourgeoise - 92800 PUTEAUX
Tél : (1) 42 04 00 88 - Fax : (1) 42 04 69 80

PC TECHNOLOGIE

BITS PER SECOND

Service lecteurs n° 22 (dernière page)

LES APPORTS DE dBASE IV

FICHE PRODUIT

- Logiciel : dBASE IV 1.5 US
- Editeur : Borland
- Prix : 8 450 F HT
  995 F HT la mise à jour depuis la version 1.1.

A la différence des précédentes versions, la «mise à jour» est un produit complet avec les manuels et le jeu de disquettes.

s'adressant aux utilisateurs (dBase IV 1.1) et l'autre aux développeurs (dBase IV 1.1 édition avancée). Il n'existe plus désormais qu'un seul produit avec tous les outils de programmation évolués, tels que le langage de création de modèles pour la génération d'applications ou encore le débogueur symbolique qui permet de corriger le source d'une application dans une fenêtre tout en le testant dans une autre. Enfin, dBase IV 1.5 possède un runtime pour la diffusion d'applications, sans que les éditeurs aient à payer des royalties à Borland. Ce dernier continuera à proposer à son catalogue dBase III Plus, dBase IV «Server Edition» ainsi que les versions de dBase IV multi-plateforme (VMS/UNIX SUN).

Plus rapide, plus sûr

dBase IV 1.5 dispose d'un gestionnaire de mémoire cache — en mémoire étendue et/ou paginée — appelé «HyperDisk» et automatiquement pris en compte lors du chargement du logiciel. Son rôle est de réduire les temps d'accès aux enregistrements le plus souvent traités.

Si cette fonctionnalité n'est pas nouvelle (présente dans la version 1.1), les performances du gestionnaire de mémoire ont, quant à elles, été améliorées : rapidité augmentée, support de Windows 3.0 et du DOS 5.0 et compatibilité assurée avec les autres programmes résidents. Les tests que nous avons effectués sur une base de plus de 15 000 enregistrements sont concluants : la version 1.5 est plus rapide que la 1.1. Recherche d'enregistrements, extraction de données, réorganisation de la base et indexation ne souffrent aucune remarque.

Cette nette amélioration se traduit également au sein du programme, soit pour l'affichage, soit lorsque l'on passe d'un module à l'autre; par exemple, de la modification des données au QBE.

Des efforts ont été portés sur la sécurité d'emploi, le concepteur s'assurant de fournir un produit fiable tant pour la manipulation courante que pour l'intégrité des données. Hélas! une coupure de courant ou l'appui sur <CTRL ALT DEL> n'offre toujours pas l'écriture sur disque des données dernièrement modifiées!

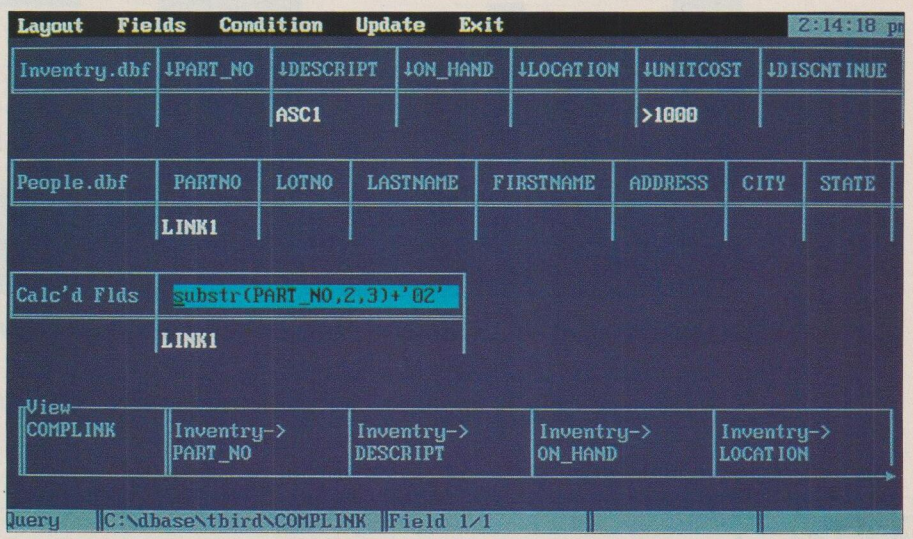
En environnement réseau, dBase IV 1.5 offre plus de garantie au travers d'un meilleur contrôle d'accès au réseau. Ainsi, la réservation automatique d'un enregistrement prévient la corruption de ses données dès lors que plusieurs utilisateurs tentent de le modifier au même moment. Dans ce contexte, le SGBD dispose des fonctions de détection automatique de modification et de rafraîchissement d'écran afin d'afficher sur tous les postes «demandeurs» la dernière mise à jour de l'enregistrement.

Enfin, dBase IV 1.5 offre un système de protection d'accès aux informations par mot de passe et dispose d'une procédure d'encryptage des données pour éviter toute consultation frauduleuse.

L'environnement Windows n'est pas oublié : Borland livre les fichiers PIF (fichiers d'information permettant de lancer une application standard à partir de Windows 3) et les fichiers d'icônes des principaux programmes constituant dBase IV. Si la procédure automatique d'installation détecte la présence de Windows 3, elle crée un groupe de programmes dBase.

Le dBase du futur

dBase IV 1.5 s'inscrit pleinement dans la nouvelle architecture de développement de Borland Boca (Borland Object Component Architecture). Cette dernière permettra à une application bureautique (dBase, Paradox, Object



dBase IV 1.5 comporte un optimiseur qui facilite le QBE dans le filtre de requête.

Vision, Quattro Pro...) de gérer des tables, des extractions, des formulaires ou des rapports sans avoir à se soucier du format des données. Ce tour de force est réalisé grâce à un noyau commun appelé «inter base» dont le code fut initialement intégré dans Object Vision.

Ubiquité

Toutes les applications fonctionnant sous Windows utiliseront ce noyau pour accéder aux données. Cette architecture présente un autre intérêt, celui de la mise à jour du noyau; toutes les améliorations qui lui sont apportées profitent instantanément à l'ensemble des applications. Ce qui n'est pas le cas avec les filtres d'importation et d'exportation.

Enfin, le noyau — qui peut être installé au sein d'une architecture client-serveur — possède des possibilités proches d'un moteur SQL, dont l'intégrité référentielle. Cette fonctionnalité sera respectée quel que soit le produit utilisant le noyau.

Les rapides corrections effectuées par Borland sur l'héritage dBase qu'Ashton-Tate lui a laissé font de ce nouveau produit davantage une version de transition (comme le montre son numéro) qu'une release majeure.

Dans le courant de 1993, la version Windows de dBase IV devrait voir le jour. Le principe de fonctionnement sera celui de la version 1.5 de dBase IV; l'investissement en formation sera ainsi conservé. Outre les opérations standards, l'utilisateur pourra définir graphiquement les liens entre les fichiers et dessiner directement à l'écran des formulaires de saisie ou de rapports. ❐

dBASE EN QUELQUES CHIFFRES

Part de marché de dBase III Plus et de dBase IV :
Etats-Unis : 47%, France : 35 à 40%
Base installée de dBase III Plus et de dBase IV dans le monde :
+ 4 millions, en France : 250 000
(Sources SPA 1990/1991)
Remarque : ces chiffres ne concernent que dBase en environnement micro. Les ventes dans l'environnement multi-plate-forme et dans l'architecture client-serveur ne sont pas ici prises en compte.

SÉLECTIONNÉE PAR **MICRO SIGMA**

LA LIBRAIRIE GRAPHIQUE **HALO Professional**

RÉPOND AUX EXIGENCES DE VOS APPLICATIONS GRAPHIQUES

- Support de nombreux **périphériques** (écran, imprimante, table traçante, souris...)
  Intégration de leurs évolutions successives
  Utilisation optimale de leurs fonctionnalités
- Disponibilité pour les **langages** C, Pascal, Fortran, Basic (mode réel), Assembleur, Ada.
- Utilisation en DOS mode réel et **mode protégé** grâce à la compatibilité avec les DOS Extenders (Rational System, Phar Lap, Ergo).
  Allocation dynamique de la mémoire : pas de module résidant, chargement du driver nécessaire lors du run-time.
  Rapidité d'exécution.
- Nombreuses fonctionnalités graphiques :
  - Diversité des primitives graphiques
  - Fenêtrage avec clipping
  - Polices de caractères vectorielles et bit map avec éditeurs
- Support technique en France assuré par **MICRO SIGMA**, distributeur agréé Media Cybernetics
- A partir de 3 950 Frs H.T.

Consultez-nous au 46 22 99 88 Fax : 47 63 84 42
Recevez une **documentation** par **FAXFACTS** au 47 63 59 93 - HALO = N° 317

MICRO SIGMA Outils de Développement
42, rue Legendre - 75017 PARIS

HALO Professional est une marque déposée de MEDIA CYBERNETICS

Service lecteurs n° 23 (dernière page)

THÉORIE

LES ALGORITHMES DE TRI

Optimiser l'accès aux données

ERIC DE KERVALEC

De nombreuses applications tirent leur intérêt de la capacité de l'ordinateur à ordonner rapidement des données initialement saisies dans un ordre quelconque. Ainsi, dans le domaine de la gestion de fichiers, un tri permet, par exemple, d'effectuer des calculs uniquement sur les enregistrements qui satisfont certains critères (comme les factures comprises entre deux dates).

De même, une procédure de tri est souvent appelée pour imprimer des rapports qui montrent des données classées par dates, par ordre alphabétique ou par montants de chiffres d'affaires.

L'intérêt de disposer les données dans un ordre déterminé n'est pas seulement esthétique : pour un être humain comme pour un ordinateur, la recherche d'une information s'effectue beaucoup plus rapidement si les données sont présentées dans un ordre qui correspond aux besoins de la recherche. Pour s'en convaincre, il suffit d'imaginer la difficulté de recherche de la définition d'un mot dans un dictionnaire si ce dernier n'était pas ordonné par ordre alphabétique.

La seule solution rigoureuse (pour un humain comme pour un ordinateur) serait d'examiner les termes un par un jusqu'à trouver celui que l'on recherche. Sur un dictionnaire de 10 000 mots, une telle recherche séquentielle n'aboutirait qu'après avoir examiné en moyenne 5 000 mots (et il faudrait passer en revue l'ensemble du dictionnaire pour un mot inconnu !).

En revanche, à partir d'un dictionnaire trié par ordre alphabétique, une simple recherche binaire (ou par dichotomie) permettrait de découvrir le mot recherché en 15 essais au maximum (log 10 000/log 2+1).

Des clés et des enregistrements

En empruntant une terminologie familière aux développeurs d'applications de gestion de fichiers, nous considérerons ici qu'un algorithme de tri doit ordonner un ensemble d'enregistrements sur la base d'un critère de tri appelé «la clé». Chaque enregistrement peut contenir beaucoup d'autres données en plus de la clé qui lui est associée. A partir d'un ensemble d'enregistrements présentés dans un ordre quelconque, l'algorithme de tri doit produire un nouvel ensemble dans lequel les clés de chaque enregistrement se suivent dans l'ordre souhaité.

On utilise généralement l'ordre alphabétique ascendant pour les clés composées de chaînes de caractères et l'ordre numérique croissant si les clés sont numériques.

Les dates peuvent, quant à elles, être converties en nombres entiers (nombre de jours écoulés depuis une date choisie comme origine dans le passé, par exemple le 1er janvier 1900) ou triées comme des chaînes de caractères si on les stocke sous la forme AAAA/MM/JJ.

Il est toutefois possible de généraliser les algorithmes présentés pour qu'ils opèrent sur d'autres types de données, à condition d'écrire la fonction de comparaison correspondante (celle-ci peut même être passée en argument lors de l'appel de la fonction de tri).

Il faut également noter que les méthodes exposées sont des algorithmes de tri interne : cela signifie qu'elles considèrent que toutes les clés sont directement accessibles sans perte de temps, quelle que soit leur position dans l'ensemble des enregistrements à trier.

Cette condition n'est généralement vérifiée que lorsque les données à trier sont présentes en mémoire. En revanche, les méthodes de tri externe sont obligatoires lorsque l'on doit trier des fichiers de taille importante (qu'il est impensable de charger entièrement en mémoire). La performance d'un tel tri étant généralement limitée par le positionnement dans le fichier et l'accès aux données, ces algorithmes accèdent au fichier de manière purement séquentielle ou au moins par blocs de données consécutifs (comme procède également le tri par fusion de listes).

Trier une structure de données est une opération fondamentale pour un ordinateur. La performance de nombreuses applications dépend de la méthode employée. Cet article présente les algorithmes les plus utilisés dans ce domaine, des plus simples jusqu'aux plus élaborés.

Pour faciliter la compréhension des algorithmes exposés dans le cadre de cet article, leur présentation suppose que l'on souhaite simplement trier par ordre croissant un tableau comportant N entiers.

La valeur d'un élément du tableau représente donc à la fois la clé et les données d'un enregistrement, et ce sont les entiers que l'on permute au sein du tableau (les exemples utilisent quant à eux des caractères de l'alphabet pour montrer le fonctionnement des algorithmes par une méthode plus «visuelle»).

Il faut toutefois noter que, dans une application réelle, les clés et les données sont souvent traitées séparément : chaque élément du tableau serait alors une structure (struct en langage C ou record en pascal) dans laquelle on ne considérerait que la clé lors des comparaisons. Il suffit pour cela de remplacer dans le code des algorithmes les références à «tab[i]» par «tab[i].cle».

Lorsque chaque enregistrement comporte beaucoup de données, il peut devenir intéressant de ne plus déplacer les enregistrements eux-mêmes mais plutôt de manipuler un tableau de pointeurs sur les enregistrements (de manière similaire aux fameux index de fichiers).

L'algorithme de tri ne permute alors que les pointeurs associés à chaque clé et les enregistrements eux-mêmes ne sont pas déplacés. Encore une fois, l'adaptation des algorithmes présentés ci-après à une telle situation est aisée lorsque leur principe a été assimilé.

Le tri à bulles

Le tri à bulles (Bubble Sort) est remarquable pour deux raisons : d'une part, cette méthode figure en introduction de presque tous les recueils d'algorithmes de tri, d'autre part, sa performance est nettement inférieure à celles des autres méthodes, sans que l'algorithme lui-même soit pour autant plus simple à comprendre ou à coder (par rapport au tri par insertion, par exemple). Cette méthode n'est donc exposée ici que pour mémoire, sans entrer dans les détails de son fonctionnement.

Le tri à bulles effectue des itérations à travers le tableau à trier, en échangeant à chaque étape les éléments adjacents dont l'ordre est incorrect.

A la fin de la première itération, l'élément dont la clé est la plus élevée se retrouve à la fin du tableau (au rang N), car il a été sans cesse échangé avec l'élément situé à sa droite.

Après le passage suivant, l'élément de rang N-1 est également à sa place, et ainsi de suite jusqu'à ce que tout le tableau soit trié.

Le nom de cet algorithme provient du fait que chaque élément remonte progressivement vers sa destination finale, de manière similaire à des bulles dans une coupe de champagne (où l'imagination des programmeurs ne va-t-elle pas se nicher !).

Malgré sa simplicité, l'algorithme du tri à bulles n'est presque jamais retenu pour des applications réelles, en raison de ses mauvaises performances dès que le nombre d'éléments à trier dépasse la centaine.

Le tri à bulles présente en théorie une performance moyenne proportionnelle à N^2, donc similaire à celle du tri par insertion qui est présenté ci-dessous; toutefois, ce dernier s'exécute environ trois fois plus rapidement dans la plupart des cas *(voir tableau comparatif)* et n'est guère plus compliqué à coder que le tri à bulles.

```
void tri_bulles( int tab[], int N )
{
     int i, j, temp;
     for( i = N; i >= 1; i– )
     {
    /* après chaque itération,
     * l'élément de rang i est à sa place
     */
         for( j = 2; j <= i; j++ )
         {
             if( tab[ j-1 ] > tab[ j ] )
             {
```

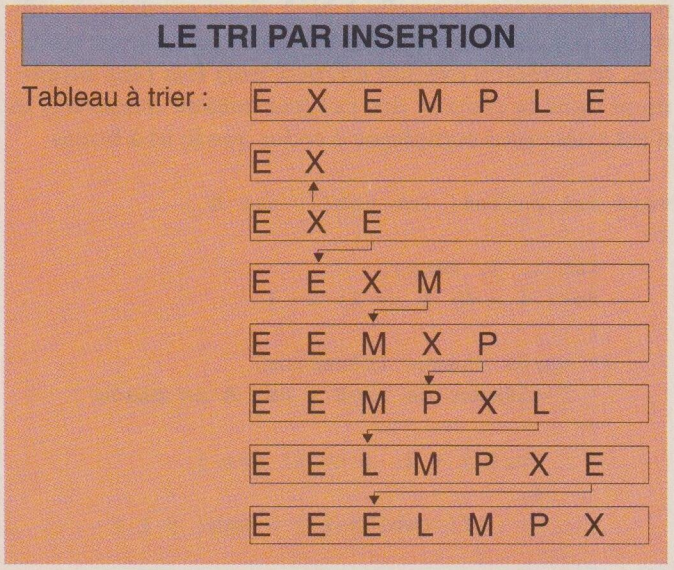
```
        /* échange de tab[j] et tab[j-1] */
                temp = tab[ j ];
                tab[ j ] = tab[ j-1 ];
                tab[ j-1 ] = temp;
            }
        }
    }
}
```

Le tri par insertion

La méthode de tri par insertion procède de manière similaire à un joueur de cartes qui ordonne son jeu : le tableau est parcouru vers le haut en insérant chaque élément à sa place parmi les éléments précédents (qui ont déjà été triés), et cela jusqu'à ce que l'ensemble du tableau ait été examiné. L'emplacement de l'élément à insérer est libéré en déplaçant vers le haut du tableau les éléments qui se trouvent entre le point d'insertion et la position initiale de l'élément examiné.

Cette méthode présente l'avantage d'être particulièrement intuitive et facile à suivre «à la main». A titre d'exemple, supposons que l'on applique la méthode de tri par insertion à la chaîne de caractères : «exemple». Les deux premiers éléments du tableau («e» et «x») sont déjà dans le bon ordre. Le troisième élément («e») doit ensuite être inséré avant celui qui précède, ce qui donne : «eexmple».

De même, le «m» et le «p» sont insérés successivement avant le «x» : «eempxle». Le «l» est ensuite inséré avant le «m», ce qui donne : «eelmpxe». Enfin, le dernier «e» est inséré avant le «l» et le tableau est alors entièrement ordonné : «eeelmpx».



Le principe du tri par insertion est donc de déplacer un index i vers la fin du tableau de manière à ce que les éléments de rang inférieur à i soient toujours ordonnés. Pour insérer l'élément de rang i à sa place, il suffit de déplacer un par un vers la fin du tableau tous les éléments dont la clé est supérieure à celle de l'élément considéré (en progressant vers le début du tableau).

Lorsque l'on rencontre un élément dont la clé est inférieure ou égale à celle de l'élément à insérer (ou le début du tableau), il n'y a plus qu'à stocker ce dernier à l'emplacement libéré par le dernier élément décalé. Le code qui implémente cet algorithme est donné ci-après :

```
void tri_insertion( int tab[], int N )
 {
     int i, j, cle;
     tab[0] = CLE_MINIMUM; /* Elément
                            * sentinelle */
     for( i = 2; i <= N; i++ )
     {
     /* Après chaque itération, les
      * éléments tab[ 1 ] à tab[ i ] sont
      * ordonnés. */
         j = i;
       cle = tab[ i ]; /* L'élément à insérer */
       while( tab[ j-1 ] > cle
 )
         {
         /* Décalage des éléments déjà
          * ordonnés */
             tab[ j ] = tab[ j-1 ];
             j--;
         }
         tab[ j ] = cle;    /* Insertion de
                             * l'élément */
     }
 }
```

Il convient de remarquer que l'élément de rang zéro du tableau est initialisé avec une valeur «sentinelle» qui doit correspondre à la plus petite des clés possibles. Cette technique améliore la vitesse d'exécution en évitant de devoir vérifier à chaque itération lors du décalage si la variable j reste bien supérieure à 1 (dans la boucle «while(tab[j-1] > cle)».

Le temps nécessaire pour trier un tableau de N éléments par insertion est proportionnel à N^2, mais il faut noter que le tri par insertion présente des performances intéressantes lorsque le tableau est déjà partiellement ordonné.

Or ce cas se produit en pratique dans de nombreuses applications de gestion de fichiers, lorsque seul le nombre d'enregistrements modifiés entre deux tris est faible par rapport au nombre total d'enregistrements.

Dans le cas idéal où le tableau est déjà ordonné, le tri par insertion présente, en effet, un temps d'exécution proportionnel à N.

Le tri ShellSort

L'algorithme ShellSort (du nom de M. Shell, son inventeur) accélère la performance du tri par insertion en autorisant les échanges entre des éléments éloignés du tableau.

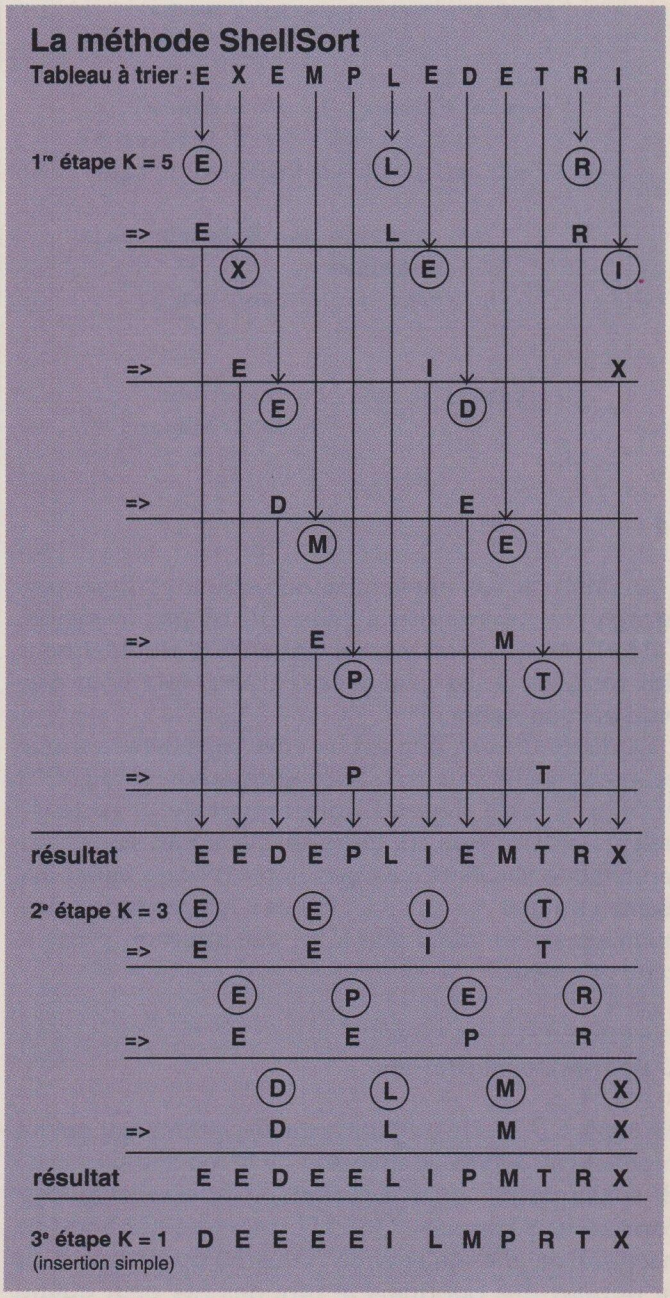
En effet, le tri par insertion ne déplace un élément vers son emplacement définitif qu'à l'aide d'échanges avec les éléments adjacents. Si le plus petit élément est initialement situé à la fin du tableau, il faudra donc décaler tous les autres éléments pour l'insérer à sa place (lors de la dernière itération), ce qui diminue fatalement la performance de l'algorithme (le pire des cas étant celui qui se produit lorsque le tableau est initialement ordonné à l'envers).

Pour éviter cet écueil, l'algorithme ShellSort réarrange le tableau en appliquant un tri par insertion à plusieurs sous-ensembles du tableau.

Ces sous-ensembles sont constitués en considérant les éléments du tableau situés à un certain intervalle k les uns des autres. A titre d'exemple, avec k = 3, on considérerait les sous-ensembles formés des éléments du tableau de rang (1, 4, 7, 10...), ceux de rang (2, 5, 8, 11...), et enfin ceux de rang (3, 6, 9, 12...). L'algorithme ShellSort trie chacun de ces sous-ensembles par insertion puis diminue la valeur de k jusqu'à terminer par k = 1 (ceci correspond à un tri par insertion «classique» et garantit donc que le tableau final est parfaitement ordonné).

La performance de la méthode ShellSort dépend de la suite des valeurs décroissantes que prend le paramètre k. Cette suite doit impérativement se terminer par la valeur 1 : k peut prendre par exemple les valeurs N/3, N/9, N/27...1.

De nombreux essais ont été effectués pour déterminer les meilleures valeurs à adopter, mais la solution optimale n'est pas connue de manière évidente (on sait en revanche que certaines suites donnent d'assez mauvais résultats, comme par exemple : 64, 32, 16, 8, 4, 2, 1). L'implémentation présentée ci-dessous utilise la suite définie par k(0)=N+1 et k(n)=(int)((5*k(n-1)-1)/11); le terme k(0) n'est pas utilisé et l'on passe à k=1 dès que k devient inférieur à 5. Pour N = 100, k prend donc les valeurs suivantes : 45, 20, 9, 4, 1.




```
void tri_shellsort( int tab[], int N )
 {
     int i, j, k, cle;
          for( k = N+1; k >1; /* rien */ )
 {
     /* Détermination de k */
     if( k < 5 )
     {
         k = 1;
     }
     else
     {
         k = (5*k-1)/11;
     }
    /* Tri par insertion sur les
     * sous-ensembles d'ordre k
     */
```

```
        for( i = k+1; i <= N; i += k )
        {
           j = i;
           cle = tab[ i ]; /* L'élément
                           * à insérer */
          while( j > k && tab[j-k] > cle)
             {
             /* Décalage des éléments déjà
              * ordonnés */
               tab[ j ] = tab[ j-k ];
                j -= k;
             }
          tab[j] = cle; /* Insertion
                        * de l'élément */
      }
   }
}
```

En raison de son implémentation simple et de ses performances supérieures à celles du tri par insertion, l'algorithme ShellSort est souvent un bon candidat pour des routines de tri générales (par exemple pour une bibliothèque runtime).

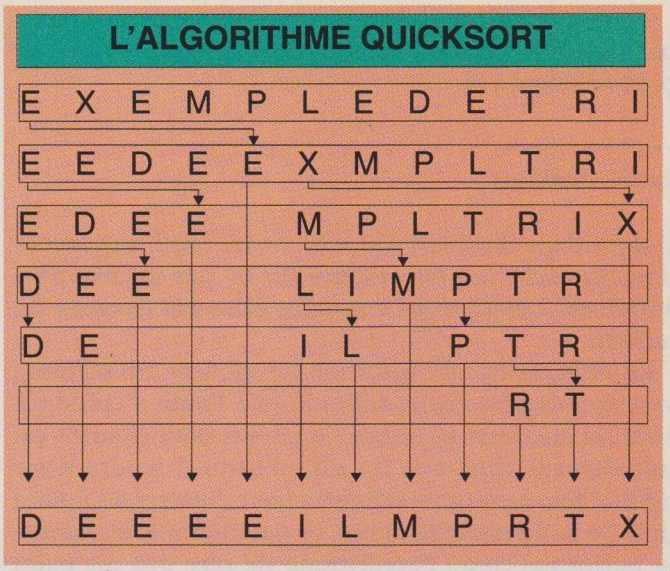
La durée d'exécution est en effet considérée comme proportionnelle à un facteur compris entre N*(log N)^2 et N^3/2 (cela dépend notamment de la suite de valeurs utilisée pour k). En revanche, il faut savoir que la méthode ShellSort n'est pas stable (l'ordre initial des éléments ayant des clés identiques n'est pas nécessairement préservé), alors que le tri par insertion possède cette propriété.

L'algorithme QuickSort

En 1960, C.A.R. Hoare publia un algorithme qui devint rapidement populaire en raison de la relative simplicité de son implémentation et surtout de ses excellentes performances : appelée «QuickSort», cette méthode ne nécessite en effet qu'environ N*log(N) opérations dans la plupart des cas (mais le pire des cas peut quand même demander jusqu'à N^2 opérations si le tableau est déjà ordonné). L'algorithme QuickSort fonctionne de manière récursive : le tableau à trier est séparé en deux parties qui sont ensuite ordonnées de manière indépendante.

Ce partitionnement se fait en sélectionnant un élément du tableau et en formant ensuite une partie qui contient les clés inférieures à celle de l'élément, et une autre qui comporte les clés supérieures à celui-ci.

L'élément autour duquel s'opère la partition se trouve donc situé à son emplacement final entre les sous-ensembles déterminés à l'étape précédente. On applique ensuite la même procédure à chacune des



deux parties situées de part et d'autre de l'élément de partition. La structure générale de l'algorithme est donc la suivante :

```
void tri_quicksort( int tab [], int premier,
int dernier )
 {
     int m;
   if( dernier > premier )
     {
        m = partition( premier, dernier );
        tri_quicksort( tab, premier, m-1 );
        tri_quicksort( tab, m+1, dernier );
     }
 }
```

La fonction tri_quicksort donnée ci-dessus ordonne la partie du tableau tab qui est délimitée par les indices premier et dernier. Pour trier l'ensemble du tableau, on effectue donc l'appel suivant :

```
tri_quicksort( 1, N );
```

Le schéma de fonctionnement donné ci-dessus fait ressortir le rôle essentiel de la fonction de partition. En effet, celle-ci doit sélectionner un élément du tableau et l'amener à sa place finale (le rang m), tout en déplaçant vers les rangs 1 à m-1 les éléments dont la clé est inférieure à celle de l'élément sélectionné, d'une part, et vers les rangs m+1 à N les éléments dont la clé est supérieure à celle de l'élément sélectionné, d'autre part.

Une stratégie générale pour parvenir à ce résultat consiste à choisir arbitrairement comme élément de partition celui situé au premier rang du tableau, c'est-à-dire l'élément tab[premier] (avec les conventions utili-

sées ci-dessus dans la définition de la fonction tri_quicksort).

On balaie ensuite le tableau avec un indice i, en garantissant à chaque itération que les conditions tab[premier+1..m] < tab[premier] et tab[m+1..i-1] >= tab[premier] sont vérifiées. Lorsque l'on inspecte l'élément de rang i, deux cas sont à considérer. Si tab[i] >= tab[premier], les conditions ci-dessus sont toujours vérifiées.

Dans le cas contraire, on échange l'élément de rang i avec l'élément de rang m+1 et on incrémente m, ce qui rétablit à nouveau les conditions de base. Lorsque l'on est arrivé à la fin du tableau, il suffit alors d'échanger tab[premier] avec tab[m] pour que la partition soit conforme aux spécifications demandées (à savoir tab[premier..m-1] <= tab[m] <= tab[m+1..dernier]). Voici donc le code de l'algorithme complet :

```
void tri_quicksort( int tab [], int premier,
int dernier )
 {
     int i, m, temp, cle;
    if( premier < dernier )
     {
     /* -- Procédure de partition -- */
         cle = tab[ premier ];
           /* L'élément de partition */
           m = premier;
     for(i = premier+1; i <= dernier; i++)
         {
         /* A chaque itération, on a
          * tab[ premier+1..m ] < cle
          * et tab[ m+1..i-1 ] >= cle
          */
             if( tab[ i ] < cle )
             {
                 m++;
                 temp = tab[ i ];
                 tab[ i ] = tab[ m ];
                 tab[ m ] = temp;
             }
         }
         temp = tab[ premier ];
         tab[ premier ] = tab[ m ];
         tab[ m ] = temp;
    /* On a maintenant :
     * tab[ premier..m-1 ] <= tab[ m ]
     * et tab[ m ] <= tab[ m+1..dernier ]
     */
    /* -- Tri récursif sur les deux
     * partitions -- */
        tri_quicksort( tab, premier, m-1 );
        tri_quicksort( tab, m+1, dernier );
     }
}
```

Cet algorithme offre, dans le cas général, des performances bien supérieures à celles des méthodes simples présentées jusqu'ici, car il ne demande en effet que N*log(N) opérations pour trier un tableau de N éléments disposés aléatoirement (*voir tableau comparatif*). Toutefois, quelques améliorations peuvent encore être apportées à l'algorithme QuickSort de base pour contourner certains de ses points faibles.

On remarque tout d'abord que la méthode utilisée pour partitionner le tableau dégénère lorsque ce dernier est déjà partiellement ordonné (car on ne bénéficie plus de l'effet réducteur d'une division par deux du nombre d'éléments à trier lors de chaque appel récursif).

Pour s'en convaincre, il suffit de considérer le cas d'un tableau entièrement ordonné : à chaque appel

Méthode de tri	Nbre d'éléments du tableau (répartition aléatoire) 500	1 000	2 000	5 000	10 000
Tri bulles	0,60	2,40	9,50	59,71	***
Tri insertion	0,27	0,88	3,52	21,97	***
Tri ShellSort	0,24	0,81	3,23	20,11	***
Tri QuickSort	0,03	0,06	0,12	0,36	0,84
Tri HeapSort	0,04	0,08	0,19	0,55	1,21

Ce tableau donne les résultats de la comparaison des divers algorithmes. Les données fournies correspondent au temps en secondes nécessaire pour trier un tableau rempli par des entiers aléatoires compris entre 1 et la taille du tableau.

LES SGBD ET LES ALGORITHMES DE TRI

Il semblerait logique, à première vue, que certains SGBD utilisent des algorithmes de tri dérivés de ceux qui sont présentés dans cet article, par exemple lorsqu'ils doivent trier une base de données pour présenter des informations dans un certain ordre.

La réalité est plus complexe : les algorithmes étudiés ici supposent en effet que les données à trier sont facilement accessibles et que c'est le nombre de comparaisons ou d'échanges qui conditionne la performance des opérations.

Or ce cas de figure ne se produit que si l'ensemble des clés que l'on trie est présent en mémoire.

Dès que cette condition cesse d'être vérifiée, c'est le temps d'accès à la mémoire de masse qui limite les performances du système. Les SGBD appliquent donc d'autres méthodes qui tiennent compte du fait que seule une fraction des clés à étudier peut être chargée en mémoire à un instant donné. Ces méthodes ne servent d'ailleurs pas uniquement à trier les données, mais surtout à permettre de retrouver rapidement l'adresse sur le disque dur de l'enregistrement associé à une clé donnée.

On désigne ces algorithmes sous le nom générique de méthodes d'indexation.

récursif «tri_quicksort(tab, m+1, dernier)», on aurait m = premier+1, ce qui entraînerait un temps d'exécution proportionnel à N^2 et N appels récursifs (d'où un risque de débordement de la pile du système). On note que cette dégénérescence de l'algorithme de partition se produit également lorsque le tableau contient de nombreuses clés identiques.

Pour diminuer le risque de dégénérescence en cas de tableau partiellement ordonné (ce qui se produit fréquemment dans la pratique), il est préférable de sélectionner au hasard l'élément autour duquel s'effectue la partition dans le tableau à trier.

La méthode la plus générale consiste à déterminer un nombre aléatoire dont la valeur est comprise entre les bornes du tableau, puis à échanger le premier élément du tableau avec celui situé au rang indiqué (cet échange a lieu juste avant d'appeler la procédure de partition).

Une technique plus simple n'utilise pas de générateur de nombres aléatoires mais échange le premier élément du tableau avec un élément déterminé arbitrairement avant de commencer le partitionnement : on peut, par exemple, considérer l'élément tab[premier + (int)((dernier-premier)/7)] comme un bon candidat à l'échange.

Une autre manière d'améliorer l'algorithme QuickSort provient du constat qu'une grande partie du temps est passée à trier des sous-tableaux de petite taille. Or l'algorithme QuickSort n'est pas très performant sur de petits tableaux et gagne à être remplacé dans ce cas par une méthode plus simple, par exemple un tri par insertion (qui n'encombre pas la pile système avec des appels récursifs). Si l'on souhaite appliquer le tri par insertion aux tableaux de taille inférieure à 15 éléments, il suffit de modifier l'implémentation de la manière suivante :

```
/* ... début de tri_quicksort ... */
 if( premier < dernier )
 {
     if( (dernier-premier) <= 15 )
     {
                    tri_insertion( tab, premier,
                                          dernier );
          /* NOTE : ici, tri_insertion ne
           * prend plus les même paramètres
           * que la version de base...
           */
     }
     else
     /*
 ... suite du tri_quicksort
 ... */
 }
```

Une méthode encore plus simple que la précédente consiste tout simplement à ne pas appliquer l'algorithme QuickSort aux sous-tableaux dont la taille est inférieure à 15 éléments. Pour cela, on remplace simplement la ligne :

```
if( premier < dernier )
```

par la ligne :

```
if( dernier-premier > 15 )
```

A l'issue de l'appel «tri_quicksort(tab, 1, N)», on obtient donc un tableau presque ordonné, sur lequel un tri par insertion s'exécute avec de bonnes performances. Pour finir le tri, il suffit donc d'appeler la fonction «tri_insertion(tab, N)».

Notons enfin que l'on peut diminuer la taille nécessaire sur la pile en remplaçant le deuxième appel récursif à la fonction «tri_quicksort(tab, m+1, dernier)» par une boucle, ce qui conduit au schéma suivant :

```
/* ... début de tri_quicksort ... */
 while( premier < dernier )
 {
     /* ... procédure de partition ... */
   tri_quicksort( tab, premier, m-1 );
   premier = m+1;
 }
```

La mise en œuvre simultanée de toutes ces améliorations (choix arbitraire de l'élément de partition, limite des appels récursifs sur les petits tableaux, remplacement de la récursion finale par une boucle) réduit considérablement le risque de dégénérescence de l'algorithme QuickSort et diminue le nombre d'appels récursifs, sans pour autant avoir un impact négatif sur les performances.

L'algorithme HeapSort

La méthode HeapSort procure des performances similaires à l'algorithme QuickSort, tout en évitant les risques liés à ce dernier : en effet, l'algorithme HeapSort ne fait pas appel à la récursivité et il garantit un nombre d'opérations proportionnel à N*log(N) quelle que soit la forme des données initiales (que le tableau soit rempli aléatoirement, déjà ordonné, ou même trié en ordre contraire).

La contrepartie de cette sécurité est une durée d'exécution presque deux fois plus longue que celle de QuickSort dans le cas de données aléatoires (*voir le tableau comparatif*), mais la vitesse d'exécution d'un tri HeapSort reste tout de même très supérieure à celles des algorithmes «simples» (tri à bulles, tri par insertion ou ShellSort).

L'algorithme HeapSort fait appel à une structure de données particulière, que l'on appelle un «tas» (heap). Un tas est en réalité un arbre binaire complet dont les nœuds contiennent les clés associées aux éléments à trier.

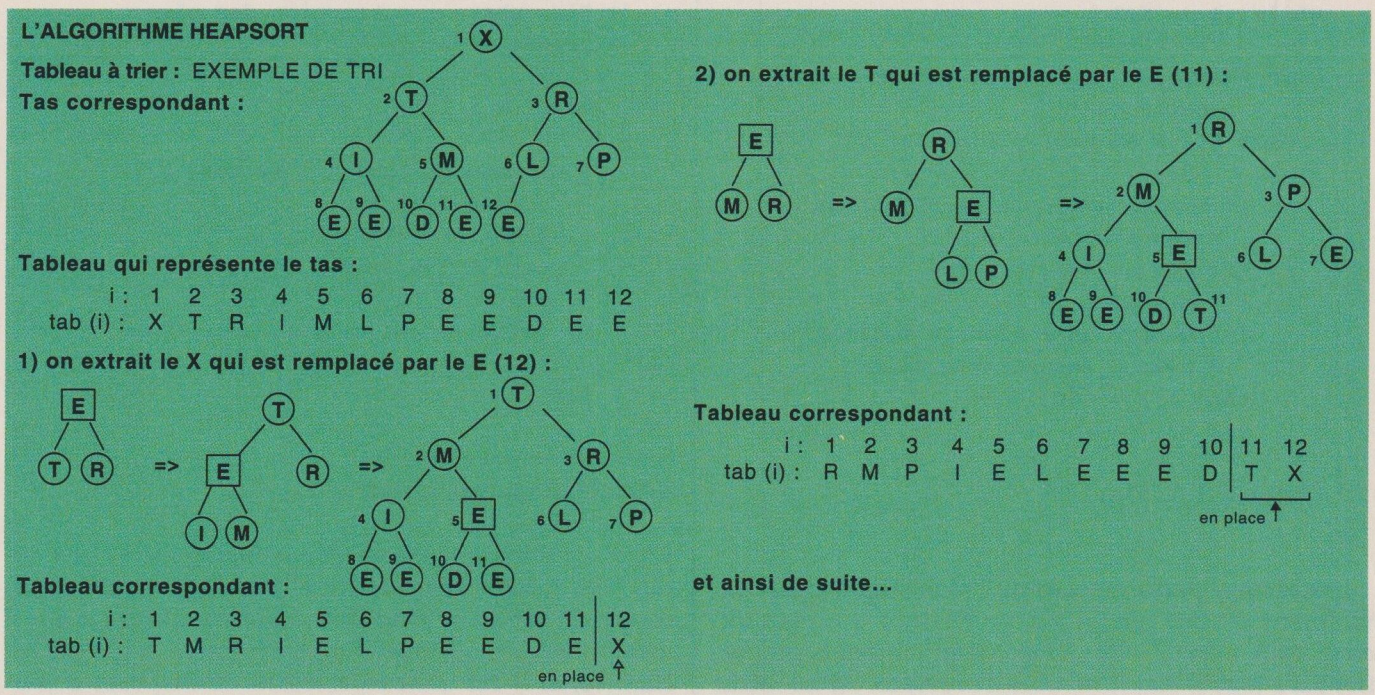
Un tas vérifie impérativement la propriété suivante : «La clé contenue dans un nœud quelconque est supérieure ou égale aux clés de ses enfants éventuels.» Le fait de parler d'un arbre binaire «complet» signifie que les nœuds sont occupés de gauche à droite pour chaque niveau de l'arbre et que l'on ne commence à remplir un niveau donné que lorsque tous les nœuds du niveau précédent sont occupés.

Contrairement à un arbre binaire général, un arbre binaire complet peut aisément être représenté par un simple tableau, dont les éléments correspondent aux nœuds de l'arbre lorsqu'on démarre à la racine et qu'on le traverse en «largeur d'abord». Ceci revient à attribuer le rang 1 à la racine, les rangs 2 et 3 à ses enfants (de gauche à droite), les rangs 4 à 7 aux nœuds du troisième niveau (toujours de gauche à droite), et ainsi de suite.

Outre sa compacité, la représentation d'un arbre binaire complet sous forme de tableau permet d'implémenter aisément les opérations nécessaires au parcours de l'arbre : si l'on considère l'élément de rang i du tableau, son parent est situé au rang (int)(i/2) et ses deux enfants sont les éléments de rangs 2*i et 2*(i+1).

Lorsqu'un tas est représenté sous forme de tableau, la condition de base à satisfaire s'écrit donc : «tab[(int)(i/2)] >= tab[i]» pour tous les rangs i entre 2 et N.

L'algorithme HeapSort fonctionne en deux temps. La première phase de l'algorithme construit un tas à partir du tableau initial, ce qui permet à la deuxième phase d'effectuer le tri en extrayant de manière répétitive le plus grand élément du tas. Cette deuxième phase ne nécessite pas de tableau auxiliaire : chaque fois que le



plus grand élément est extrait du tas, la taille de ce dernier diminue et libère ainsi la place nécessaire pour stocker dans le tableau initial l'élément que l'on vient d'extraire...

L'algorithme HeapSort est construit autour d'une fonction dont le but est de réorganiser un tas dont la valeur d'un élément a été diminuée, ce qui risque d'entraîner une violation de la propriété de base («tout nœud possède une clé supérieure ou égale à celle de ses enfants éventuels»). La fonction verifie_tas examine donc le tas qui résulte de la modification et procède, le cas échéant, à des échanges pour que le nouveau tas vérifie encore la condition fondamentale. Pour «réparer» le tas, la fonction verifie_tas examine les enfants de l'élément modifié; si la valeur de ce dernier est inférieure à celle de l'un de ses enfants (ce qui viole le principe de base), l'enfant dont la valeur est la plus élevée est échangé avec l'élément modifié. On examine alors les enfants qui dépendent du nœud où l'on vient de déplacer l'élément modifié, et l'on procède, si nécessaire, à un nouvel échange dans les mêmes conditions que pour l'opération précédente. Cette procédure est répétée en progressant vers le bas de l'arbre binaire tant que la propriété fondamentale n'est pas vérifiée et que l'on n'a pas atteint le bas de l'arbre.

Voici donc le code de la fonction verifie_tas (l'argument k indiquant le rang du nœud qui a été modifié) :

```
void verifie_tas( int tab[], int N, int k)
 {
    int j, cle;
    cle = tab[ k ]; /* L'élément modifié */
    while( k <= N/2 )
     {       j = k+k;
      /* Choix de l'enfant de valeur
        * supérieure */
       if( j < N && tab[ j ] < tab[ j+1 ] )
             j++;
        /* Vérification de la propriété
          * fondamentale */
        if( cle >= tab[ j ] )
            break;
        /* Echange de l'élément modifié
          * avec l'enfant examiné */
        tab[ k ] = tab[ j ];
        k = j;
     }
     tab[ k ] = cle;
}
```

Les bonnes performances de l'algorithme HeapSort viennent du fait que la réorganisation d'un tas qui contient N éléments ne demande qu'un temps proportionnel à log(N) (car le nombre de niveaux est lui-même proportionnel à log(N)). L'insertion d'un élément dans un tas ou l'extraction de son plus grand élément nécessitent donc un nombre d'opérations proportionnel à log(N).

La fonction tri_heapsort utilise la fonction verifie_tas pour construire le tas à partir du bas, en considérant chaque élément du tableau initial comme la racine d'un petit tas (on démarre en réalité au milieu du tableau, car les éléments situés au-delà de la moitié du tableau correspondraient à des tas n'ayant qu'un seul élément, sur lesquels la fonction verifie_tas n'effectuerait aucun travail).

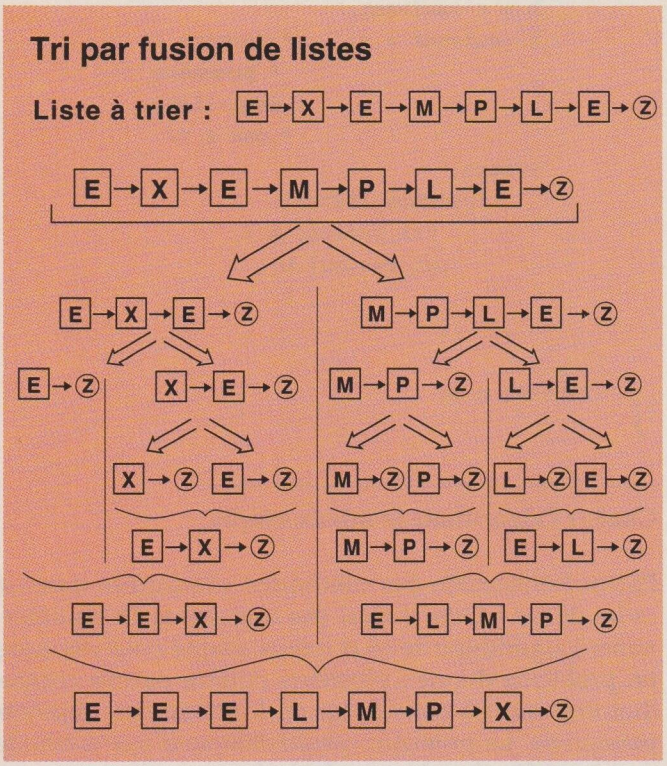
Une fois que l'ensemble du tableau a été inséré dans le tas, l'algorithme HeapSort extrait à plusieurs reprises l'élément le plus grand du tas et remplit ainsi le tableau à partir de la fin. Il faut noter que l'élément le plus grand est toujours situé à la racine du tas, c'est-à-dire au rang 1 du tableau. Pour procéder à son extraction, on échange ce premier élément du tableau (le plus grand) avec le dernier élément du tas (situé au rang N du tableau), on décrémente ensuite N, et on appelle à nouveau la fonction verifie_tas. Le fait de décrémenter N diminue la taille du tas sur lequel travaille verifie_tas, ce qui garantit que la partie «haute» du tableau n'est plus modifiée.

Voici donc le code correspondant à l'algorithme HeapSort proprement dit :

```
void tri_heapsort( int tab[], int N )
 {
     int k, temp;
   /* 1ère phase : construction du tas. */
     for( k = N/2; k>=1; k– )
         verifie_tas( tab, N, k );
   /* 2ème phase : extraction des plus
    * grands éléments.
    */
     while( N > 1 )
     {
     /* Echange de tab[ 1 ] et tab[ N ]. */
         temp = tab[ 1 ];
         tab[ 1 ] = tab[ N ];
         tab[ N ] = temp;
         N–;
         verifie_tas( tab, N, 1 );
     }
 }
```

Trier des listes par fusion

Les algorithmes présentés plus haut supposent que les données à trier soient représentées en mémoire sous forme de tableau. Toutefois, certaines applications manipulent des informations représentées par une liste chaînée, et il est donc intéressant de disposer d'une



méthode de tri adaptée à cette structure de données. L'algorithme de tri par fusion (MergeSort) est tout à fait indiqué dans ce contexte.

Il faut noter que le tri par fusion présente des performances similaires à celles de la méthode HeapSort, avec en particulier une durée d'exécution proportionnelle à N*log(N) quel que soit l'ordre initial des données à trier.

Comme son nom l'indique, l'opération fondamentale du tri par fusion est constituée par la fusion de deux listes ordonnées, à l'issue de quoi on obtient une liste ordonnée qui contient les éléments des deux listes initiales.

La construction de la nouvelle liste à partir des listes initiales (que nous appellerons liste_a et liste_b) s'effectue élément par élément : si la clé de l'élément courant de liste_a est inférieure à celle de l'élément courant de liste_b, on ajoute à la liste fusionnée l'élément courant de liste_a et on passe à l'élément suivant de cette liste (tant qu'on n'en a pas atteint la fin); le cas contraire (liste_b->cle < liste_a->cle) se traite de manière tout à fait symétrique.

Dans la réalité, on manipule directement les pointeurs des listes initiales pour éviter de devoir allouer de la mémoire pour contenir le résultat de la fusion. Le tri a donc lieu «sur place», de même qu'avec les autres méthodes qui opèrent sur un tableau d'entiers. Les listes manipulées par notre algorithme sont définies de la manière suivante :

```
struct elem
     {
         int cle;
         struct elem * suivant;
     };
```

Comme pour le tri par insertion, l'implémentation de l'algorithme de fusion est facilitée si les listes qui lui sont passées en argument se terminent par un élément «sentinelle» dont la clé est supérieure à toutes les valeurs possibles et qui pointe sur lui-même. Cet élément est appelé «z» et supposé avoir été initialisé comme suit (dans une variable globale au module) :

```
struct elem * z;
/* ... initialisation de z ... */
 z = (struct elem *) malloc( sizeof( *z ));
 z->cle = CLE_MAXIMUM;
 z->suivant = z;
```

La fusion est obtenue en manipulant les pointeurs des listes passées en arguments à la fonction fusionne_listes. Le programme appelant est donc supposé avoir alloué la mémoire correspondante et initialisé ces listes avec les valeurs à trier. Avec ces conventions, voici comment s'écrit l'algorithme de fusion :

```
struct elem * fusionne_listes( struct elem * a,
struct elem * b )
 {
     struct elem * c;      /* Pointe sur l'élément
                            * courant de la liste
                            * qui résulte de la
                            * fusion */
      c = z;
     do
     {
         if( a->cle <= b->cle )
         {
         /* Ajoute à la liste l'élément courant
          * de a, fait pointer c sur cet élément,
          * avance l'élément courant de a.
          */
             c->suivant = a;
             c = a;
             a = a->suivant;
         }
         else
         {
         /* Ajoute à la liste l'élément courant
          * de b, fait pointer c sur cet élément,
          * avance l'élément courant de b.
          */
             c->suivant = b;
```

```
            c = b;
            b = b->suivant;
        }
    }
    while( c != z );
    /* c pointe sur l'élément courant de la
     * liste qui résulte de la fusion, z pointe
     * sur son premier élément.
     */
    c = z->suivant;     /* Récupère le début de
                         * la liste. */
    z->suivant = z;     /* z pointe à nouveau sur
                         * lui-même. */
    return c;
}
```

L'algorithme tri_fusion utilise la fonction fusionne_listes pour trier de manière récursive la liste qui lui est passée en argument. Le principe de cet algorithme ressemble à celui de la méthode QuickSort : après avoir divisé la liste à trier en deux parties, on trie (récursivement) chacune des parties, puis on les fusionne pour obtenir la liste complète correctement ordonnée.

La récursivité s'arrête lorsqu'une des deux listes à fusionner ne comporte qu'un seul élément : celui-ci est directement inséré à sa place par l'appel à la fonction fusionne_listes.

Il faut noter que l'ensemble du tri par fusion s'opère en réorganisant les pointeurs de la liste initiale (puisque la fonction fusionne_listes effectue son travail en manipulant les pointeurs des listes qui lui sont passées en argument). Voici donc le code de la fonction tri_fusion :

```
struct elem * tri_fusion( struct elem*c )
{
    struct elem * a, * b;
    if( c->suivant != z )
    {
      a = c;
    /* a pointe sur le début
                * de la liste */
      b = c->suivant->suivant->suivant;
        while( b != z )
        {
                /* c avance d'un élément
                 * quand b avance de
                 * deux éléments ; quand b
                 * pointe sur la fin
                 * de la liste, c pointe
                 * sur le milieu */
            c = c->suivant;
            b = b->suivant->suivant;
        }
        b = c->suivant;
        c->suivant = z;    /* Termine la
                            * première
                           * demi-liste
                            * par z */
        return(
            fusionne_listes(
                tri_fusion( a ),
               tri_fusion( b )
                           )
              );
    }
    return( c );
}
```

Choisir l'algorithme le mieux adapté

Parmi l'ensemble des méthodes présentées dans le cadre de cet article, il n'est pas toujours facile de déterminer l'algorithme de tri le mieux adapté pour résoudre un problème donné : plusieurs critères parfois contradictoires sont à considérer en fonction des objectifs poursuivis. La première caractéristique d'une méthode de tri est l'ordre de grandeur de sa performance en fonction du nombre N d'éléments à ordonner (on considère généralement que la performance d'un algorithme de tri est proportionnelle au nombre moyen de comparaisons qu'il effectue).

Les algorithmes les plus simples (tri à bulles, tri par insertion, ShellSort) ont malheureusement un temps d'exécution proportionnel à N^2, ce qui ne les rend réellement envisageables que pour trier un nombre limité d'éléments (quelques centaines ou quelques milliers au maximum) : en effet, le temps d'exécution est multiplié par 100 quand le nombre d'éléments est multiplié par 10.

En revanche, les algorithmes les plus évolués (QuickSort, HeapSort, tri par fusion de listes) se contentent d'un temps d'exécution proportionnel à N*log(N) : lorsque le nombre d'éléments à trier est multiplié par 10, le temps d'exécution n'augmente alors que d'un facteur 20.

La mémoire nécessaire peut également entrer en ligne de compte surtout si l'on envisage de trier des tableaux de taille importante : ainsi, les méthodes QuickSort et le tri des listes par fusion risquent-elles de faire déborder la pile des appels de fonctions si l'on ne prévoit pas suffisamment de place pour le pire des cas.

Les autres méthodes envisagées (notamment le HeapSort dont les performances sont proches du QuickSort) présentent en revanche l'avantage de ne pas réclamer plus de mémoire que l'espace nécessaire pour stocker le tableau à trier. Sans être comparables en termes de performances avec les méthodes les plus évo-

BIBLIOGRAPHIE

• Jon BENTLEY :
Programming pearls (Addison-Wesley, 1986).

• Jon BENTLEY :
More programming pearls (Addison-Wesley, 1988).

• G.H. GONNET & R. BAEZA-YATES :
Handbook of algorithms and data structures in Pascal and C (Addison-Wesley, 1991).

• Robert SEDGEWICK :
Algorithms (Addison-Wesley, 1983).
Algorithms in C (Addison-Wesley, 1990).

luées, on notera que le tri à bulles et le tri par insertion sont des algorithmes stables. Cela signifie que ces méthodes préservent l'ordre initial des enregistrements qui partagent la même clé.

Ainsi, lorsqu'une méthode de tri stable est appliquée à un fichier d'adresses qui se trouve déjà classé par ordre alphabétique pour l'ordonner par code postal, les enregistrements qui partagent le même code postal restent dans l'ordre qu'ils occupaient dans le fichier initial et se trouvent donc toujours classés par ordre alphabétique après le deuxième tri (par code postal). En revanche, un algorithme instable aurait totalement détruit l'ordre initial des enregistrements qui partagent la même clé : à code postal identique, l'ordre des enregistrements serait aléatoire. Toutefois, la stabilité de l'algorithme de tri n'est pas un critère de choix fondamental pour la plupart des applications. Il est en effet possible de simuler cette propriété en créant une clé composée.

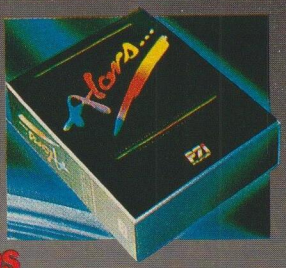
Le dernier critère à considérer pour choisir un algorithme de tri est la simplicité de son implémentation. Il existe de nombreux cas pour lesquels il n'est pas indispensable d'obtenir la meilleure performance possible : le nombre des enregistrements à trier peut rester faible (de l'ordre de quelques centaines), et la routine de tri n'être utilisée que rarement (pas plus d'une fois par mois, par exemple). Dans ces conditions, mieux vaut choisir l'algorithme le plus simple, car le temps perdu pour implémenter et déboguer un algorithme plus complexe ne sera peut-être jamais regagné lors de l'utilisation réelle du produit final...

Voilà d'ailleurs ce qui explique pourquoi un algorithme comme le tri par insertion est encore utilisé de nos jours : ne nécessitant pas d'espace mémoire supplémentaire, stable et ne demandant que quelques lignes de code simples à comprendre, il représente une solution acceptable lorsque la performance n'est pas le critère essentiel. ❐

Alors... MD

UN AGL POUR NANTUCKET 5

- Dictionnaire de données
- Générateurs de sources
- Bibliothèque



La maîtrise du temps

Alors... intègre trois modules complémentaires : un dictionnaire de données, des générateurs de sources et une bibliothèque, destinés à aider le développeur à gagner du temps et créer des sources très lisibles.
A l'image des moteurs relationnels, le dictionnaire de données permet de créer les fichiers, les index, les contrôles, les liens interfichiers et d'en garantir la cohérence hors des lignes de programmes.

Un très petit nombre de lignes à maintenir

Le dictionnaire élimine toute l'écriture des contrôles interfichiers. La bibliothèque comporte une cinquantaine de fonctions majeures pour les développeurs (attribution d'un nom de fichier temporaire en réseau, log d'erreur automatique, recherche d'un champ dans un autre fichier...). Les générateurs développent un ensemble de fonctions modulaires (entrées/sorties, menus déroulants, doc. technique...).

Une maintenance aisée

Voici quelques exemples :
Un champ change de longueur? Tous les éléments utilisant ce champ fichiers, index, écrans et variables mémoires sont modifiés en cascade et reconstitués dynamiquement.
Un index est nécessaire pour un nouvel état? Vous le déclarez dans le dictionnaire et tous les programmes déjà écrits le reconnaissent immédiatement et le prennent en compte.
Pour en savoir plus renvoyez-nous le coupon ci-dessous.

PZI DEVELOPPEMENT
88 avenue Kléber - 75116 Paris
Tél. 47.27.01.29 ■ Fax 47.04.66.08

OFFRE SPECIALE 8 900 F HT 23 140 F HT 42 900 F HT
Version Monoposte ~~15 000 F HT~~ ■ 5 p. ~~39 000 F HT~~ ■ 10 p. ~~78 000 F HT~~

Je souhaiterais ❑ Une documentation.
❑ Le manuel de découverte avec la disquette de démonstration. Ci-joint un chèque de 177,90 F TTC.
❑ Assister à une démonstration gratuite.

Nom _____ Fonction _____
Société _____ Tél. _____
Adresse _____

Nantucket est une marque déposée par Nantucket Corporation

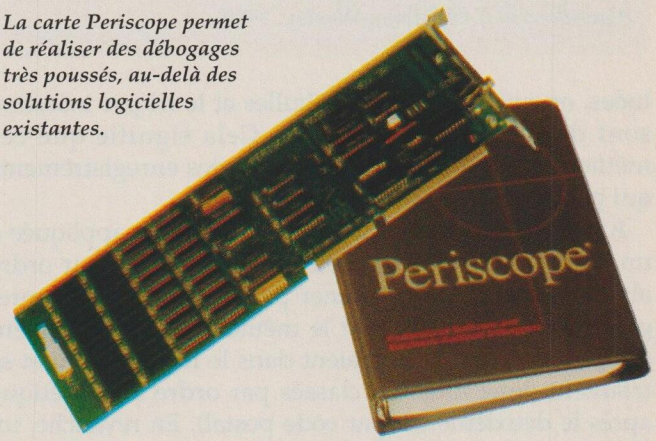
Service lecteurs n° 24 (dernière page)

DÉCOUVERTE

PERISCOPE
Le débogueur hardware

FRANÇOISBERGEON

Les produits de la gamme Periscope ne s'adressent pas à la majorité des développeurs. Nul besoin en effet d'un débogueur matériel lors de la mise au point d'un logiciel de comptabilité générale ou de facturation, par exemple. Mais dès lors que l'application effectue des appels de bas niveau, ou même des accès directs au hardware, l'utilité des débogueurs matériels est unanimement reconnue. Car rares sont les outils de développement capables d'aider à la mise au point d'un device driver, du Bios résident sur une carte d'extension ou encore de la séquence de boot d'un nouveau système d'exploitation.



La carte Periscope permet de réaliser des débogages très poussés, au-delà des solutions logicielles existantes.

Déboguer les résidents

La base du système Periscope est un débogueur symbolique logiciel, commun à tous les modèles de cartes, et pourvu des fonctions standards que l'on est en droit d'attendre d'un tel produit (affichage du code source, support de programmes destinés à Windows, utilisation d'un second écran, remote debugging, etc). De plus, de nombreuses options d'utilisation permettent, par exemple, de chronométrer l'exécution du programme, d'utiliser les registres de débogage des 386/486 ou encore de créer un «batch» contenant une série de commandes Periscope.

Bien que l'interface utilisateur propose l'émulation des touches de fonction Turbo-Débogueur ou CodeView, l'apprentissage du logiciel n'est pas aisé, et la documentation qui l'accompagne ne s'avère pas toujours très claire en ce qui concerne son utilisation. Notons que le débogueur Periscope supporte les compilateurs pascal, C et C++ Borland, C Microsoft et C JPI/TopSpeed, ainsi que les linkers Plink et RTlink.

Du fait de la nature résidente du logiciel Periscope, le programme à mettre au point doit être appelé par l'utilitaire RUN qui a pour but de charger le code exécutable en mémoire et de passer la main au débogueur. L'utilisateur aura pris soin de lancer auparavant le logiciel TS afin de générer une table des symboles employés par Periscope. Celle-ci est créée à partir des informations contenues dans un fichier .MAP lors de l'édition des liens, ou encore grâce aux informations de débogage symbolique présentes dans le fichier exécutable.

La mise au point des «device drivers», quant à elle, peut être effectuée en chargeant le logiciel Periscope à partir du CONFIG.SYS grâce à l'utilitaire SYSLOAD.SYS. Fonctionnellement identique à la commande INSTALL disponible sous MS-DOS version 3.31 ou supérieure, cet utilitaire permet en outre d'assurer le chargement d'un programme résident avant celui d'un device driver, ce qui n'est pas le cas d'INSTALL.

Par ailleurs, Periscope permet à l'utilisateur d'installer une routine spécifique qui sera appelée lors de l'exécution d'un point d'arrêt, par exemple dans le but de réinitialiser des données pour le programme en cours de mise au point.

Bouton-poussoir d'interruption

L'originalité du logiciel Periscope est d'être couplé à l'interruption 02, également appelée «NMI» (Non Maskable Interrupt). Les débogueurs logiciels tirent

Les debogueurs symboliques tels que Turbo-Debugger ou CodeView font merveille pour la majorité des applications. Dans certains cas, cependant, une solution logicielle seule ne suffit pas. Periscope propose tout une gamme de matériel avec pour base un débogueur symbolique logiciel pour permettre de dépasser les limites des débogueurs classiques.

habituellement parti de l'interruption 03 (Breakpoint) et 04 (Step) pour exécuter un programme en mode «pas à pas» ou insérer des points d'arrêt.

En remplaçant une instruction donnée par INT 03h, le débogueur s'assure ainsi la reprise du contrôle de la machine. Cette technique possède cependant une faille : si l'appel d'un sous-programme ou d'une routine de service d'interruption (ISR) provoque un blocage, le point d'arrêt n'est pas exécuté, et il ne reste plus qu'à rebooter la machine, sans que la cause du problème n'ait été identifiée. Avec le logiciel Periscope, et grâce à un bouton-poussoir relié à la ligne NMI présente sur le bus, l'utilisateur peut à tout moment suspendre l'exécution et activer le débogueur. Ce fonctionnement s'applique, par exemple, au processus de mise au point de logiciels résidents. Le bouton NMI est présent sur tous les modèles de Periscope et il représente à lui seul une grande partie de l'intérêt d'un débogueur matériel.

Le premier produit de la gamme Periscope, le modèle II, se compose du logiciel et du bouton-poussoir uniquement. Bien que le logiciel seul soit également commercialisé sous le nom de Periscope II-X ou Periscope/EM, son intérêt en tant que tel reste très réduit et son acquisition ne se justifie que dans le cas où l'utilisateur confectionne lui-même le bouton NMI.

Relogeable en mémoire haute

Une autre caractéristique singulière du programme Periscope est son aptitude à se charger, sous certaines conditions, en dehors des 640 ko gérés par MS-DOS. Le logiciel résident nécessite environ 400 ko de RAM, mais si la machine est dotée d'un microprocesseur 386 et du gestionnaire de mémoire 386Max de Qualitas, celui-ci peut être installé en mémoire étendue, n'occupant ainsi aucune place en mémoire centrale. Cette optimisation de l'utilisation du 386 n'est possible que grâce à une option d'installation de 386Max dédiée à l'utilisation de Periscope et à une version spéciale du débogueur (Periscope/EM). Il est toutefois regrettable qu'un support identique ne soit pas proposé pour d'autres gestionnaires de mémoire tout aussi répandus que 386Max comme, par exemple, QEMM-386 de Quarterdeck.

Carte de mémoire protégée

Dans le cas où le processeur n'est pas un 386/486 ou lorsque l'utilisateur se refuse à se servir de 386Max, la carte Periscope modèle I permet toutefois de reloger le logiciel en dehors de l'espace mémoire de MS-DOS. Cette carte contient de 512 ko à 2 Mo de RAM paginée à la manière des cartes EMS. L'accès à la mémoire de la carte est réalisé à l'aide d'une fenêtre de 32 ko logée entre l'adresse 640K et 1M et protégée contre toute écriture accidentelle. La mémoire présente sur la carte est destinée à recevoir à la fois le programme Periscope et ses tables de données (symboles, etc.).

Bien que la solution consistant à recourir à 386Max plutôt qu'à un Periscope modèle I sur une machine 386 soit moins onéreuse, cette dernière est néanmoins la seule à permettre le débogage des ROM Bios et des séquences de boot. En effet, le contenu de la mémoire de la carte modèle I n'est pas effacé lors d'un «short boot». Le débogueur peut ainsi être activé dès le début de l'initialisation du système, soit par appui sur le bouton-poussoir, soit par insertion de l'instruction INT 02h (NMI) ou INT 03h (Breakpoint) au sein du code à mettre au point. Une option d'installation de Periscope permet en outre de considérer la RAM présente sur la carte comme une ROM Bios dans le but de lancer automatiquement le débogueur lors du boot.

Débogueur temps réel

La carte Periscope modèle IV, haut de gamme des débogueurs matériels, est plus particulièrement destinée à la mise au point des applications temps réel. Fonctionnant

LA GAMME PERISCOPE

• **Periscope I et I/MC :** Logiciel Periscope/EM et carte de mémoire protégée contenant de 512 ko à 2 Mo de RAM. Periscope I/MC est destiné au bus MCA. Periscope I n'utilise aucune place mémoire dans les 640 ko de MS-DOS. Prix : 6 950 F HT (7 550 F HT pour Periscope I/MC), carte équipée de 512 ko. Ancienneté : 1988

• **Periscope/EM :** Version du logiciel compatible avec 386Max et BlueMax; mêmes fonctionnalités que Persicope I sur 386 ou 486. Prix: 2 950 F HT.
Ancienneté : 1991

• **Persicope II :** Bouton NMI et logiciel standard.
Prix : 1 590 F HT. Ancienneté : 1985

• **Periscope II-X :** Logiciel standard seul. Prix : 1 290 F HT.
Ancienneté : 1986

• **Periscope III (supprimé) :** Logiciel standard et carte mémoire 64 ko.

• **Periscope IV :** Logiciel Periscope/EM, carte temps réel et pod dépendant du microprocesseur (286, 386, 486, 386SX en cours de développement). Vitesse d'horloge jusqu'à 25 ou 33 MHz. Prix : de 13 950 F HT à 33 950 F HT.
Prix des pods : 7 250 F HT (286/386), 8 950 F HT (386/486)
Distributeur : Micro Sigma.
Ancienneté : 1989

sur une machine 286, 386 ou 486, la carte modèle IV est reliée à un «pod» qui vient s'intercaler entre la carte mère et le microprocesseur. Ne contenant pas elle-même de mémoire protégée, elle peut être avantageusement complétée par une carte modèle I.

Des points d'arrêt matériel

Dès lors que la carte modèle IV est installée, de nouvelles fonctions sont disponibles au sein du logiciel. Il est ainsi possible de programmer des points d'arrêt matériels permettant à Periscope de reprendre automatiquement la main lors de l'accès à une zone mémoire donnée, voire à un port d'entrée-sortie. Des points d'arrêt itératifs (après un nombre déterminé d'exécutions d'une instruction) ou des déclenchements séquentiels (après reconnaissance d'une série d'événements prédéterminés) viennent compléter un système de comparaison de données. Bien que certains de ces modes de fonctionnement soient également proposés par des débogueurs logiciels classiques, ceux-ci ont généralement pour effet de ralentir considérablement l'exécution du programme à mettre au point, rendant de ce fait leur utilisation peu efficace. Periscope emploie la carte modèle IV pour déclencher ces points d'arrêt programmés, l'exécution du programme n'est nullement altérée dans ce cas.

Par ailleurs, la mise au point d'applications temps réel est rendue possible grâce à la fonction de postanalyse d'exécution de Periscope IV. Un buffer circulaire de 16 ko situé sur la carte modèle IV permet d'enregistrer jusqu'à 2 047 événements. A chacun d'eux correspondent 80 octets d'information : outre l'adresse (sur 32 bits) de l'instruction exécutée, sont sauvegardées les données concernées par cette instruction (32 bits également), ainsi que les différents indicateurs d'état du microprocesseur et de son bus d'entrée-sortie. Il est ainsi possible de «tracer» à rebours les derniers événements qui sont survenus au niveau du microprocesseur, que ceux-ci se soient produits en fonctionnement normal ou dans le cours d'une routine de traitement d'interruption.

Les limitations

Dès lors que le logiciel Periscope est installé, le buffer circulaire est continuellement mis à jour. La cause du déclenchement d'une interruption d'exception sur 386 peut être ainsi décelée en appelant le débogueur à l'aide du bouton-poussoir, puis en analysant ce «trace buffer», par exemple. En outre, des options d'exécution de Periscope permettent de programmer un point d'arrêt lorsque le buffer circulaire est plein, ou encore de ne capturer que certains événements présélectionnés. Seule limitation de ce système : du fait qu'ils surviennent indépendamment du microprocesseur, les événements DMA ne sont pas enregistrés dans ce buffer.

Les fonctions de mise au point de Periscope IV peuvent être également utilisées en dehors de l'environnement MS-DOS, voire sur une architecture Intel 286 ou 386/486 non compatible PC. Dans ce cas, la machine supportant le programme à mettre au point est couplée à un PC qui exécute Periscope. Le «pod» connecté au microprocesseur hôte est relié à une carte Periscope modèle IV installée sur le PC de travail. Ce mode de fonctionnement, appelé «passive remote debugging», n'autorise cependant que l'analyse du «trace buffer»; bien que des points d'arrêt soient programmables sur la machine hôte, leur activation n'interrompt pas l'exécution du programme cible pour autant.

De plus, il n'est pas possible d'accéder à la mémoire ou aux ports d'entrée-sortie de la machine de tests. Un débogage à distance «actif» est néanmoins possible, mais, dans ce cas, la machine hôte doit pouvoir supporter l'installation d'un logiciel et d'une carte spécifiques

et optionnels. Notons enfin que l'utilisation de Periscope IV sur des machines à bus MCA n'est possible qu'avec l'option «active remote debugging», la carte modèle IV n'étant commercialisée qu'au format ISA (et donc compatible EISA).

Une configuration délicate

La gamme des débogueurs proposés par Periscope confronte l'utilisateur à un choix délicat. Le modèle IV, par exemple, propose des fonctionnalités très séduisantes, pour un prix qui l'est moins (de 1 895 $ à 3 695 $!). Le modèle I, quant à lui, ne se justifie que sur des machines 286 ou encore dans le but de déboguer une séquence de boot ou le Bios d'une carte. Son prix est de 595 $ pour une machine ISA/EISA et de 695 $ pour la version MCA.

Par ailleurs, différents modèles de «pods» sont disponibles pour Periscope IV, et il n'est pas toujours aisé de déterminer celui qui fonctionnera pour une configuration donnée (type de processeur, disposition de la carte mère, vitesse d'horloge, etc.). A l'heure actuelle, par exemple, le «pod» destiné au 386SX n'est pas encore commercialisé.

L'installation des cartes peut poser d'importants problèmes de compatibilité. Ainsi, il est reconnu que les cartes Periscope modèle I ne fonctionnent pas sur certaines machines Olivetti, et que, d'autre part, le bouton-poussoir ne peut pas être installé sur les machines pour lesquelles la ligne NMI est absente du bus.

Lors de nos tests, la configuration des cartes modèle I et modèle IV a été particulièrement longue et fastidieuse. Fort heureusement, la documentation relative à l'installation des différentes cartes s'avère très claire, et une pince d'extraction du microprocesseur est même livrée avec le «pod» du modèle IV. Cette pince est destinée à aider l'utilisateur dans cette phase de mise en place particulièrement délicate qui consiste à intercaler le «pod» entre le processeur et la carte mère (extraction du processeur, insertion du «pod» en lieu et place, puis insertion du processeur sur le «pod»).

Mais ne nous y trompons pas : la difficulté d'installation et de configuration des produits Periscope n'est pas due à un manque de finition de la part du fabricant. Elle est inhérente à la nature des débogueurs dont le fonctionnement est extrêmement proche du hardware.

Utilisation courante

A l'usage, Periscope se révèle très efficace pour mettre au point les programmes jusqu'alors réputés «indébogables»... L'utilisation couplée avec 386Max ou une carte de mémoire protégée permet, en effet, de travailler sur des applications de grande taille, tout en conservant la quasi-certitude qu'aucun blocage irrémédiable du système ne surviendra : l'accès au débogueur peut se faire à tout moment grâce au bouton-poussoir, l'abandon du programme cible et le retour vers MS-DOS étant généralement possibles à partir du logiciel.

Par ailleurs, la nature résidente de Periscope permet au développeur de lancer une session de mise au point sans pour autant l'avoir préméditée. En prenant de plus l'habitude de compiler le programme en cours de tests avec les options adéquates (voire l'exécution de l'utilitaire TS dans le fichier Make), l'affichage du code source et des symboles peut être obtenu dès l'appui sur le bouton-poussoir.

Le logiciel Periscope permet d'afficher une zone mémoire en fonction d'une structure de données prédéterminée, par exemple un PSP (Program Segment Prefix). Par contre, cette structure doit être déclarée au préalable pour être utilisable.

Contrairement à des débogueurs plus spécialisés, Periscope ne permet pas l'affichage du contenu d'une structure C ou d'une classe C++. Cet inconvénient découle de son caractère généraliste et de sa compatibilité avec de nombreux compilateurs.

D'ailleurs, même si le langage C et le C++ sont sup-

LES UTILITAIRES DE PÉRISCOPE

CLEARNMI.COM	Démasquage régulier de NMI.
CONFIG.COM	Configuration du logiciel Periscope;
INT.COM	Affichage des vecteurs d'interruption.
PS.COM	Logiciel débogueur résident.
PSKEY.COM	Sélection des «hot-key» pour activer Periscope.
PSTERM.COM	Programme de communication pour «remote debugging».
PSTEST.COM	Programme de test Periscope I.
PS4TEST.EXE	Programme de test Periscope IV.
RS.COM	Définition des structures de données et alias.
RUN.COM	Chargement et exécution d'un programme sous le débogueur.
SKIP21.COM	Résident pour désactiver Periscope IV lors d'une INT 21.
SYMLOAD.COM	Chargement des tables de symboles en mémoire.
SYSLOAD.SYS	Chargement d'un programme .COM dans CONFIG.SYS.
TS.COM	Génération des tables de symboles.
USEREXIT.COM	Exemple de routine utilisateur exécutée lors d'un point d'arrêt.
WAITING.COM	Programme d'avant-plan pour libérer le «DOS busy flag».

portés, Periscope est plus particulièrement axé sur le développement en assembleur du fait de ses possibilités techniques avancées.

La mise au point de routines de traitement d'interruptions est, effectivement, grandement facilitée par les fonctionnalités spécifiques du modèle IV. Mais le «trace buffer» peut être utilisé dans bien d'autres cas. Il rend, par exemple, possible l'analyse du fonctionnement de certains programmes en partant d'un état final connu; l'outil idéal du «reverse engineering» en quelque sorte...

Ce buffer circulaire disponible avec Periscope IV est très certainement la fonctionnalité la plus marquante des différents produits de la gamme et tend même à dévaloriser les autres matériels proposés.

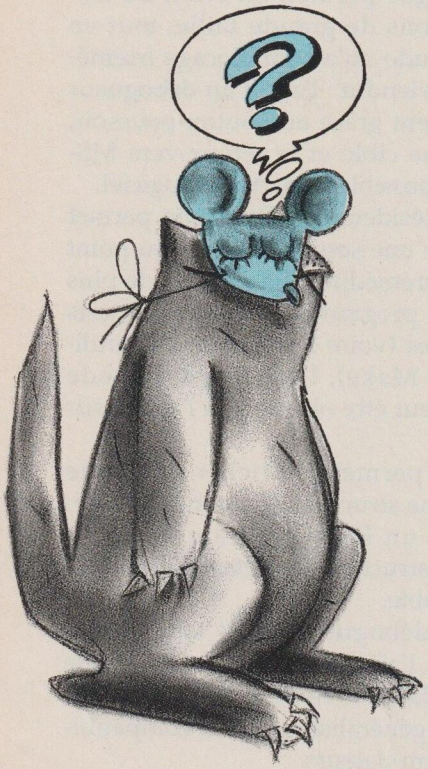
Un prix d'entrée élevé

Le prix élevé de ce modèle, bien que compréhensible, découragera certainement de nombreux acheteurs potentiels, de même que l'apprentissage approfondi, indispensable pour profiter au maximum des possibilités offertes. L'investissement financier que représente un système Periscope opérationnel fait que ces outils restent réservés à une minorité de développeurs pour laquelle ces produits sont rééllement rentables : les concepteurs de TSR, de device drivers, de systèmes d'exploitation, d'environnements multitâches, de Bios ou d'application temps réel, par exemple. N'oublions pas qu'une configuration Periscope modèle IV complète revient, en effet, au même prix qu'une machine 386 ou 486 correctement équipée.

The Periscope Company
1197 Peachtree St. Plaza Level
Atlanta, GA 30361 U.S.A
Tel: (19-1) (404) 875-8080
Fax: (19-1) (404) 872-1973

En France :
Micro Sigma
42, rue Legendre, 75017 Paris
Tel: (1) 46.22.99.88
Fax: (1) 47.63.84.42

Mal masquer n'est pas jouer !



PANEL PLUS II est un gestionnaire de masques d'écran

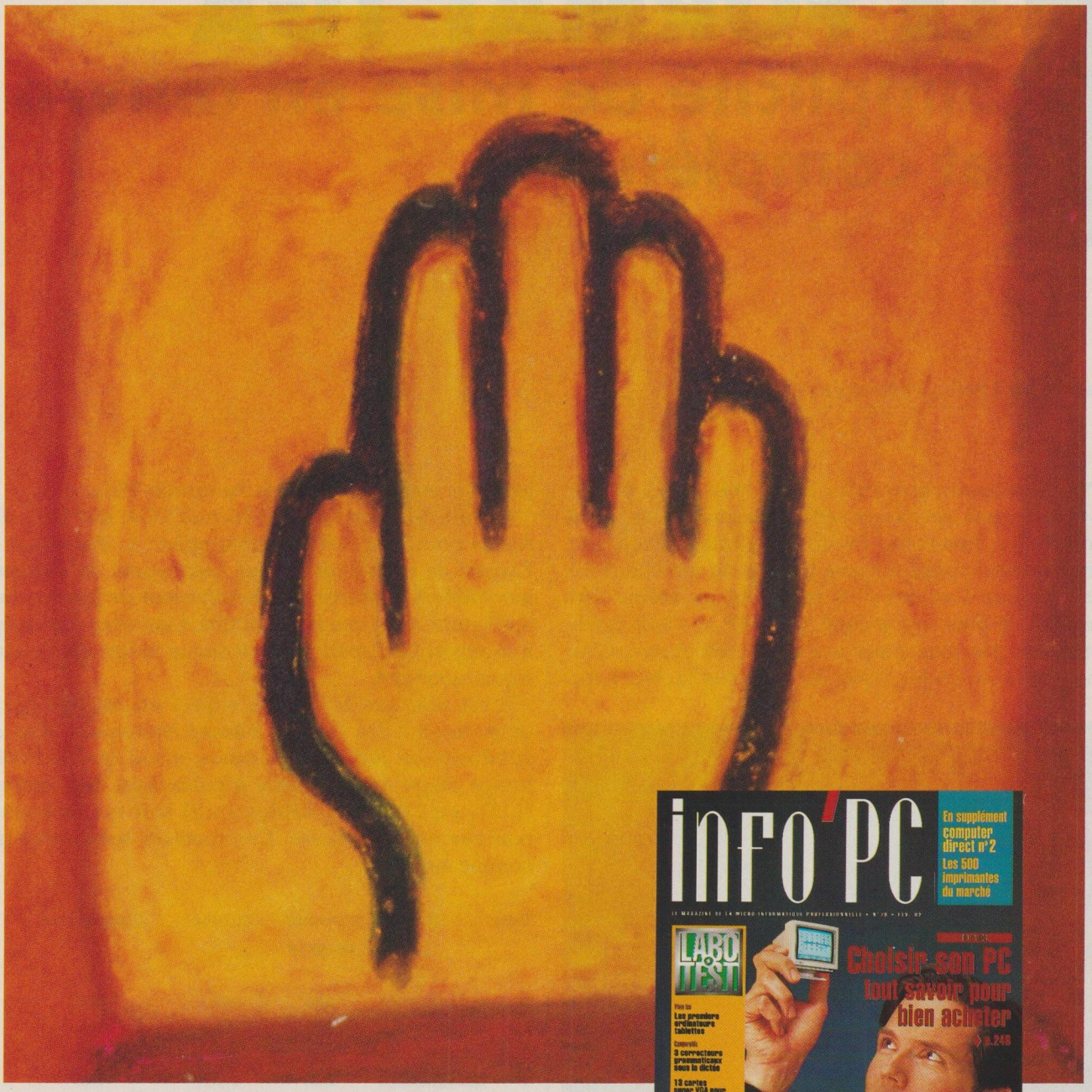
- **Portable** sur de nombreux systèmes grâce aux sources des librairies C
- Disponible sous DOS, OS/2, UNIX, VMS... et même Windows 3.0
- Conforme au standard CUA /SAA
- PANEL PLUS II **comprend** essentiellement :
  - un outil interactif de création de masques d'écran (wysiwyg)
  - un générateur de code source en langage C
  - une librairie de fonctions qui permet de gérer et de manipuler les masques
- Les masques peuvent inclure des fenêtres, des menus hiérarchiques, des zones déroulantes éditables, des écrans virtuels, des messages d'aide, des contrôles de validation...
- Support technique en France assuré par **Micro Sigma**, distributeur agréé Roundhill Computer Systems Limited
- A partir de 3 950 Frs H.T., libre de droits de diffusion

Micro Sigma Outils de Développement
42, rue Legendre - 75017 PARIS

Consultez-nous au 46 22 99 88 Fax : 47 63 84 42
Recevez une **documentation** par **FAXFACTS** au 47 63 59 93 - N° 821

Service lecteurs n° 25 (dernière page)

PANEL Plus II est une marque déposée de Roundhill Computer Systems Limited



TOUCHEZ l'innovation du bout des doigts.

Aujourd'hui, les choix en micro-informatique passent forcément par INFO PC. Un véritable outil d'exigence et de rigueur.

Toutes ces questions que vous, professionnels et utilisateurs, pouvez vous poser sur un logiciel ou un matériel micro-informatique, INFO PC y répond de façon pertinente et sûre.

Capacités techniques des produits, fiabilité des matériels, pérennité des fournisseurs, ouverture des logiciels n'auront pas plus de secret pour vous qu'ils n'en ont pour nous. Doté depuis 1988 d'un labo-test unique en Europe, INFO PC et ses 30 ingénieurs, ont déjà mis à l'épreuve plus de 2 500 matériels et logiciels afin de garantir à vos décisions, sécurité et objectivité.

IDG COMMUNICATIONS FRANCE

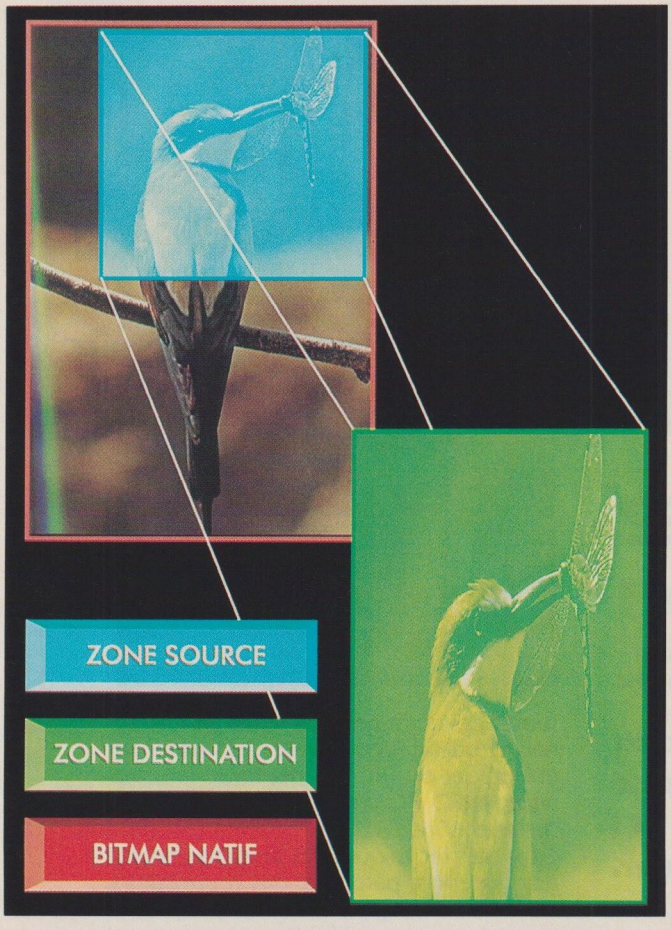
DIRECT Associés

OBJET

LE FORMAT TGA
Simplicité de mise en œuvre et complétude

CHRISTOPHE LEPECQ ET NORBERT RIMOUX

De nombreux produits sous Windows et sur Macintosh reconnaissent le format TGA, dont PhotoStyler, Image-In, Photoshop, Image Prep, etc. Parmi les principaux formats bitmap du marché, TGA se révèle être l'un des plus complets, au même titre que TIFF, le plus sophistiqué, ou BMP. Il s'agit également d'un des plus simples à mettre en œuvre. C'est sans doute la conjugaison de tous ces facteurs qui explique en partie le succès du format TGA.



Définition du «bitmap virtuel» tel qu'il sera vu par le programme d'application (zone verte).

En fait, il ne manque au format TGA que trois éléments pour être réellement complet : la définition de l'aspect de ratio d'une image (permettant de s'adapter aux différents écrans), un identificateur de formats sans ambiguïté, ainsi que la possibilité d'encoder des images à la répartition spatiale quelconque (autre que le classique rectangle).

Il est vrai que ce dernier aspect est négligé par la quasi-totalité des formats bitmap classiques. TGA est subdivisé en huit sous-formats, selon que le fichier image dispose d'une table d'indexation des couleurs (LUT, ou Look Up Table), qu'il soit monochrome, couleur ou compressé. Ces formats sont identifiés comme suit :

Type	LUT	Niveaux de gris	Couleur	Compression
1	obligatoire	non	oui	non
2	optionnelle	non	oui	non
3	optionnelle	oui	non	non
9	obligatoire	non	oui	RLC
10	optionnelle	non	oui	RLC
11	optionnelle	oui	non	RLC
32	obligatoire	non	oui	Huffman, Delta, RLC
33	optionnelle	non	oui	Huffman, Delta, RLC

L'en-tête TGA proprement dit comporte 18 octets, et sa structure est la suivante :

1 octet	Length_of_Identification (taille du commentaire)
1 octet	Cmap_Type (1 ou 0 : présence ou non d'une LUT)
1 octet	Image_Type (type d'image selon tableau ci-dessus)
5 octets	Cmap_Spec (caractéristiqes de la LUT)
10 octets	TGA_Image_Spec (caractéristiques de l'image)

De plus en plus fréquemment, on retrouve l'extension TGA dans les options d'importation et d'exportation des logiciels graphiques travaillant sur des images bitmap. Peu connu du grand public, le format TGA (contraction de Targa, les contrôleurs graphiques de TrueVision au travers desquels il s'est fait connaître) a été exploité initialement dans les principaux logiciels graphiques professionnels de l'univers MS-DOS : Tips, RIO, Topas, Lumena, Mirage, Crystal, etc.

A cet en-tête s'ajoutent deux champs de longueur variable :

- le commentaire, ou identificateur d'images, dont la taille est définie par le membre Length_of_Identification et ne peut dépasser 255 caractères (ce champ peut être vide).
- la table d'indexation des couleurs, dont la présence est indiquée par Cmap-Type et la taille par Cmap_Spec *(voir plus loin)*.

Les données proprement dites commencent à la fin de la LUT lorsqu'elle est présente. Aussi est-il laissé à l'application le soin de calculer l'offset du début d'image. L'opération reste très simple et consiste à ajouter l'en-tête (18 octets), le champ Length_of_Identification et la taille de la LUT (fournie par les trois champs de Cmap_Spec).

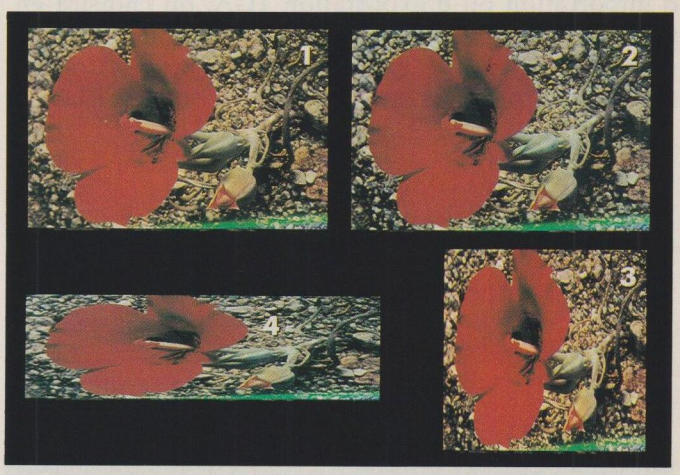
Avant de procéder à une description plus détaillée de chaque champ, on peut donc récapituler la structure de l'en-tête TGA en définissant comme suit une structure en langage C :

```
typedef struct
{
      unsigned int          Origin;
      unsigned int          Length;
      unsigned char         Depth;
}TGA_Cmap_Spec;
      (structure de la LUT)

typedef struct
{
      unsigned Attribute    :4;
      unsigned Reserved     :1;
      unsigned Origin       :1;
      unsigned Interleave   :2;
}TGA_Image_Descriptor;
      (descripteur de l'image)

typedef struct
{
      unsigned int          X_Org;
      unsigned int          Y_Org;
      unsigned int          Width;
      unsigned int          Height;
      unsigned char         Depth;
      TGA_Image_Descriptor  Descriptor;
}TGA_Image_Spec;
      (caractéristiques de l'image)

typedef struct
}
      unsigned char
      Length_of_Identification;
      unsigned char         Cmap_Type;
      unsigned char         Image_Type;
      TGA_Cmap_Spec         Cmap;
      TGA_Image_Spec Image;
}Entete_TGA;                (en-tête TGA)
```



1- Original (1 024 x 688)
2- Luminosité -10 rouge, +10 vert, + 20 bleu
Définition 1 024 x 688
Traitement inactif sur les valeurs 50 à 100 de la LUT
3- Contraste 30, définition 512 x 512
4- Luminosité -10 rouge, +10 vert, +20 bleu
Tous les index de la liste sont actifs
Définition 1 024 x 300.

LE FORMAT TGA

Une structure X de type Entete_TGA pourra être définie selon les modalités classiques du langage C, afin d'interpréter le contenu de l'en-tête TGA.

Signification des champs

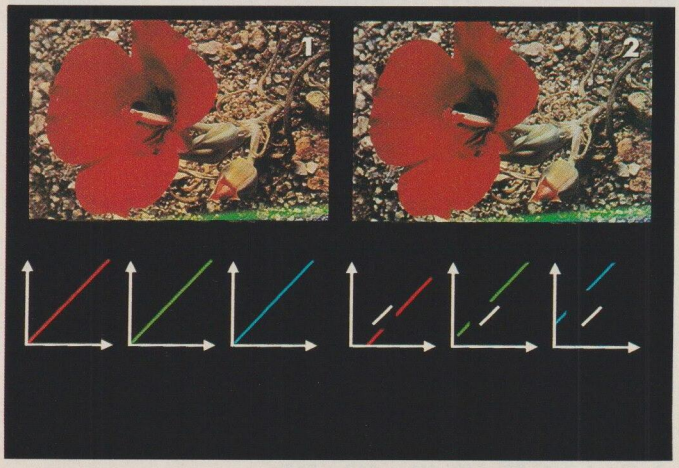
Nous avons déjà évoqué brièvement les trois premiers champs de l'en-tête du format TGA.
Le premier (X.Length_of_Identification) et le deuxième (X.Cmap_Type) indiquent respectivement la taille du commentaire lié à l'image et si le fichier est pourvu d'une table d'indexation des couleurs (LUT).
Quant au troisième (X.Image_Type), c'est lui qui va permettre aux applications de reconnaître rapidement la nature de l'image grâce à ses principales caractéristiques (couleur, monochrome, compression, etc.).

Comme on a pu le voir plus haut, huit «sous-formats» TGA sont ainsi définis. Le plus répandu est sans doute le type 2 qui permet d'encoder une image de qualité photographique sans compression.

On a ainsi le choix entre une profondeur de couleur de 16 bits (32 768 couleurs, 1 bit alpha), 24 bits (16 millions de couleurs) ou 32 bits (16 millions, 8 bits alpha).

Le format TGA type 2, s'il est bien adapté pour le chargement et la sauvegarde rapide d'images couleurs haute résolution, ne se montre pas idéal pour l'archivage étant donné l'absence de compression.

Lorsque l'on ne désire réaliser qu'un seul filtre d'importation ou d'exportation TGA, on choisira en tout cas le type 2, qui est incontestablement le plus universel des huit (et éventuellement le type 10, similaire à ceci près qu'il intègre une compression RLC).



1- Original, réponse linéaire par couleur fondamentale
2- Après traitement de luminosité -10, 10, 20; les valeurs de 500 à 100 ne sont pas actives (partie blanche des couches RGB).

X.Cmap.Origin : il s'agit de la position du premier index de couleurs, permettant à une application de se positionner directement sur le début de la LUT.
Bien que redondante, cette information peut être utile pour vérifier si une application n'a pas utilisé l'espace réservé à la LUT pour y insérer d'autres informations, comme le fait Lumena, par exemple, pour le codage d'une vignette de prévisualisation.

X.Cmap_Length : nombre d'éléments constituant la LUT. Généralement 256 dans le cas d'une image codée sur 8 bits. Les autres types de codage (24 et 32 bits) n'utilisent généralement pas de LUT, dans la mesure où chaque point inclut l'intégralité de l'information sur sa couleur.
Toutefois, on peut envisager de stocker des informations utiles, telles que des tables de compensation de couleurs évitant la mise à jour intégrale d'un fichier lors d'une modification globale de contraste, de luminosité, etc.
Cette possibilité est à manier avec moult précautions et si possible au sein d'une même application, dans la mesure où elle n'est pas du tout standardisée par le format TGA. Dans ce cas de figure, la LUT serait constituée par 3 x 256 entrées (une table de 256 entrées par couleur fondamentale).

X.Cmap_Depth : nombre de bits constituant une entrée de la LUT. Il n'existe pas de limitation spéciale à ce sujet dans le format TGA. Les valeurs de 16, 24 et 32 sont toutefois les plus répandues.
Remarque : les informations contenues dans ces trois champs permettent à une application de calculer l'offset des données dans le fichier bitmap. La taille de la LUT est, en effet, fournie par la formule suivante :

```
X.Cmap.Depth/3 * X.Cmap.Length (taille en bits)
```

valeur à laquelle il suffit d'ajouter les 18 octets de l'entête, ainsi que la taille du commentaire lié à l'image (fournie par X.Length_of_Identification), pour obtenir l'offset.

X.Image.X_Org : abscisse de l'origine (coin inférieur gauche de l'image).

X.Image.Y_Org : ordonnée de l'origine (coin inférieur gauche de l'image). Remarque : ces deux valeurs permettent de mémoriser la position de l'image à l'écran et ne sont utiles que pour le stockage d'une portion de l'écran.

X.Image.Width : largeur de l'image en nombre de points. Limitée à 65 536 points, ceci excédant largement les besoins actuels en imagerie (512 à 4 096 points généralement).

Moins cher qu'un progiciel et il en contient des dizaines

Nous vous offrons gratuitement 3 disquettes exclusives WEKA, pleines de programmes d'application qui vous feront gagner du temps et vous éviteront des erreurs de saisie. Précisez sur le bon de commande le format de disquette que vous désirez (3"1/2 ou 5"1/4).

Avec le langage C sur IBM PC/PS et compatibles, vous apprendrez à maîtriser totalement les divers aspects du Langage C, jusque dans ses variantes les plus avancées.
Programmeur averti ou professionnel, il vous suffira de vous laisser guider par une méthode didactique développée par l'auteur pour pouvoir enfin réaliser tous vos projets.
Gestion de fichiers particuliers, applications graphiques, travail en réseau, portabilité vers d'autres systèmes d'exploitation, programmation des ports d'entrée/sortie... n'auront bientôt plus de secrets pour vous et vos applications personnelles ou professionnelles deviendront de plus en plus performantes.
Ouvrage de référence, le guide WEKA, grâce à son système de compléments/mises à jour vous maintiendra en permanence au top niveau de la programmation moderne.

595 F TTC Franco - Plus de 2600 pages en feuillets mobiles, présentés dans une reliure permettant d'incorporer les compléments/mises à jour.

EXTRAITS DU SOMMAIRE

Le Langage C et son environnement de programmation : • Conception du Langage C • Standards du Langage C • Environnement de programme • Eléments du Langage C • Les fonctions librairie-Library C.
La programmation du logiciel en C : • La programmation en C • Initiation à la gestion des fichiers.
Le Langage C et les systèmes d'exploitation : • C et le système d'exploitation UNIX • C et le système d'exploitation MS-DOS • C et le système d'exploitation OS/2.
L'accès au matériel au travers du langage C : • Fonctions d'aide pour programmer le Hardware des PC/PS • Création de modules assembleur.
C-Toolbox© : • Les aides à la programmation • Traitement des fichiers autres que texte • Routines de sauvegarde.
Les outils d'aide à la programmation : • Les langages C usuels • Utilisation professionnelle du Langage C. Environnement Turbo C, Turbo C++, Quick C et C Microsoft • Générateurs d'écrans et applications s'appuyant sur les langages C et C++.
Applications graphiques : • Graphique sous environnement MS-DOS.
Utilitaires, trucs et astuces : • Utilitaires portables MS-DOS ⇔ UNIX • Utilitaires destinés à améliorer le traitement des fichiers • Trucs et astuces destinés à améliorer la programmation • Utilitaires pour imprimantes • Utilitaires système.
Annexes : • Les tables indispensables en C • Règles et syntaxe • Tables de conversion.

LA GARANTIE WEKA

"Satisfait ou Remboursé"

1. Si, à reception de l'ouvrage, vous estimez qu'il ne correspond pas complètement à votre attente, vous pouvez le renvoyer sous 15 jours aux Éditions WEKA et être alors remboursé.
2. La même garantie vous est consentie pour les envois de compléments/mises à jour. Vous pouvez les interrompre à tout moment, sur simple demande écrite, ou retourner tout complément qui ne vous satisferait pas dans un délai de 15 jours après reception.

Éditions WEKA
82, rue Curial
75935 Paris cedex 19
Tél. : (1) 40 37 01 00
Télex : 210 504 F
Fax : (1) 40 37 02 17

BON DE COMMANDE

à renvoyer avec votre règlement sous enveloppe non affranchie à :
Éditions WEKA - Libre Réponse n°5 - 75941 Paris cedex 19

OUI, envoyez-moi ..... exemplaire(s) de :
"Le langage C sur IBM PC/PS et Compatibles". (Réf. 9800).
3 volumes 20 x 23 cm, plus de 2600 pages, au prix de 595 F* TTC franco.
+ mon cadeau gratuit : 3 disquettes (format ❑ 3"1/2 ou ❑ 5"1/4).
❑ Envoi par avion : + 110 F par titre.

J'ai bien noté que cet ouvrage sera actualisé et enrichi tous les 2 ou 3 mois par des compléments/mises à jour de 200 pages au prix de 336 F* TTC franco le complément. Je pourrai interrompre ce service à tout moment par simple demande écrite et bien évidemment, je bénéficie de la garantie WEKA.
Je joins mon règlement de ..... F TTC par :
❑ Chèque bancaire ❑ CCP (à l'ordre des Éditions WEKA)
Nom : ..... Prénom : .....
Société : .....
Adresse : .....
Ville : .....
Code Postal : |_|_|_|_|_| Téléphone : .....
Date : ..... Signature obligatoire :

* Prix révisables en fonction de l'évolution des tarifs de nos propres fournisseurs
Offre valable jusqu'au 30.06.92.

251806

Éditions WEKA - SARL au capital de 2 400 000 F - RC Paris B 316 224 617

Service lecteurs n° 26 (dernière page)

X.Image.Height : hauteur de l'image en nombre de lignes.

X.Image.Depth : nombre de bits nécessaires au codage d'un pixel. Généralement 8, 16, 24 ou 32.

X.Image.Descriptor.Attribute : nombre de bits d'attribut, encore appelés «bits alpha», par pixel. Habituellement 0,1 ou 8. Par exemple, un fichier 32 bits n'exige que 24 bits par pixel pour le codage des couleurs (8 bits par composante RVB). L'utilisation des 8 bits restants, les bits d'attribut, est laissée à la charge des applications. En règle générale, les bits alpha servent à coder les informations de transparence ou de masquage.

X.Image.Descriptor.Reserved : information réservée pour usage futur.

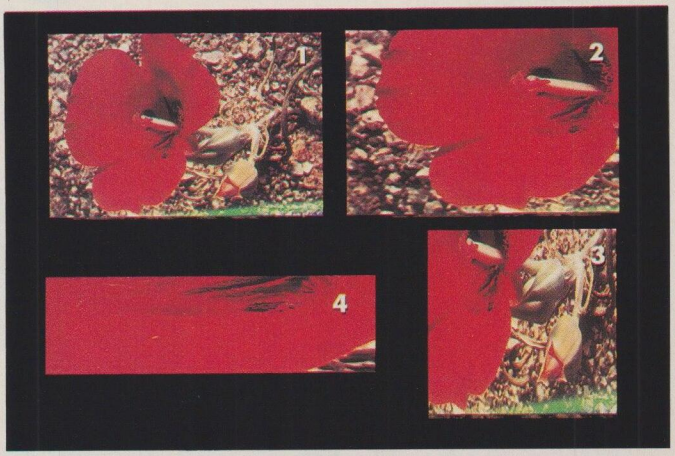
X.Image.Descriptor.Origin : renvoie la valeur 0 si l'origine de l'image est située en bas à gauche, ou 1 si elle se trouve en haut à gauche. Ce bit permet de savoir si la première ligne de l'image se situe en fin ou en début de fichier. Les spécifications ATT Truevision considèrent que la première ligne du fichier correspond à la ligne inférieure de l'écran (le bit est positionné à 0).

Ce choix paraît erroné toutefois, dans la mesure où la plupart des périphériques d'acquisition et de sortie fonctionnent selon le principe inverse : une application a donc intérêt à positionner le bit d'origine à 1, afin d'éviter de créer préalablement à toute acquisition un fichier «à vide» (valeur 0). Le bit d'origine peut servir par ailleurs à obtenir une rotation de 180° de l'image, sans pour autant avoir à manipuler les données du fichier.

X.Image.Descriptor.Interleave : description de l'entrelacement de l'image. L'immense majorité des images TGA sont codées avec une valeur 0 pour cette quantité (pas d'entrelacement). L'information n'est réellement utile que dans le cas d'adaptateurs d'écrans utilisant l'entrelacement (facteur 2 pour le CGA —> Descriptor.Interleave = 1, par exemple, ou 3 pour les cartes ATT PC —> Descriptor.Interleave = 2) et lorsque le fichier image est constitué d'une simple vidange de la mémoire sur le disque. En pratique, on préférera négliger cette information (et la positionner à 0) afin de s'affranchir des contraintes liées au matériel.

Codage et méthodes de compression

Une fois l'en-tête TGA décodé, une application dispose de toutes les informations nécessaires au chargement ou au traitement de l'image. La convention retenue pour l'ordre de lecture des informations



1- Original
2- Source (0, 0, 511, 511) destination (0, 0 ,700, 400)
3- Source (200, 0, 900, 400) destination (0, 0, 700, 400)
4- Source (300, 100, 500, 300) destination (0, 0, 512, 344).

pour chaque composante RGB est similaire à celle du format BMP, soit B, G, R, alpha.
Une structure C correspondante pourrait ainsi prendre la forme suivante :

```
typedef struct
        {
        unsigned char B;
        unsigned char G;
        unsigned char R;
        unsigned char O; }Color_32;
```

Ainsi que nous l'avons vu plus haut, les types 9, 10, 11, 32 et 33 du format TGA peuvent être compressés. Les TGA types 32 et 33 mettent en œuvre des techniques Delta et Huffman dont la description dépasse le cadre de cet article. Les types 9, 10 et 11 font appel, en revanche, à une technique simple et fort répandue, le RLE ou Run Length Encoding.

Le principe du RLC consiste à compter le nombre de pixels adjacents identiques, et à le coder sous la forme d'un paquet contenant le nombre d'occurrences et la valeur RGB du pixel. Un fichier compressé selon la méthode RLC peut contenir simultanément des paquets compressés et des paquets natifs pour lesquels la compression ne serait pas efficace.

Chaque ligne de balayage débute par un octet indiquant le nombre de pixels pris en compte. Si le bit 7 de ce compteur est à 1, le paquet est compressé.

L'application sait alors que le point suivant doit être répété autant de fois que le compteur le spécifie (-127). Si le même bit 7 se trouve à 0, le compteur indique alors quel est le nombre de points natifs à suivre.

L'encombrement de chaque point dépend, bien sûr, de la profondeur de codage de l'image. Le codage RLC

est efficace pour des images contenant de larges plages unies, mais peut se montrer carrément pénalisant pour des images digitalisées à partir de documents en «tons continus» (le fichier pouvant alors prendre davantage de place une fois compressé).

Préférer la compression indépendante

Quoi qu'il en soit, on préférera, pour des raisons de compatibilité, ne pas utiliser les méthodes de compression proposées par le format TGA lors de l'importation ou de l'exportation d'images.

Lorsque des problèmes d'encombrement se posent (stockage, transport), il est plus souhaitable d'utiliser des programmes de compression de type PKZIP ou LHARC, souvent plus efficaces et ayant l'avantage d'être indépendants de l'application qui a la charge du décodage ou de l'encodage TGA.

Avertissement : le format TGA a été créé en environnement PC.

Ce qu'il faut savoir

Adapté de l'ouvrage *Les fichiers graphiques sur ordinateur*, Christophe Lepecq et Norbert Rimoux, Editions Armand Colin / InfoPC.
Se reporter à ce livre pour la théorie du codage des images bitmap et pour les méthodes de compression Delta et Huffman.
Se reporter à l'article de Norbert Rimoux paru dans le numéro 2 de *Langages et Systèmes*, pour un exemple de transcodage TGA / TIFF en C++.

L'ordre de stockage des octets se conforme donc aux conventions Intel, soit octet de poids faible suivi de l'octet de poids fort. L'exploitation d'un fichier TGA sur une plate-forme matérielle différente, notamment Motorola/Macintosh, doit tenir compte de ce fait. Toutefois, seule la lecture des informations de l'en-tête est concernée.

Il faut alors inverser le sens des octets pour les informations de type INTEGER (2 octets) ou plus larges. ❐

PROTEGEZ VOS LOGICIELS

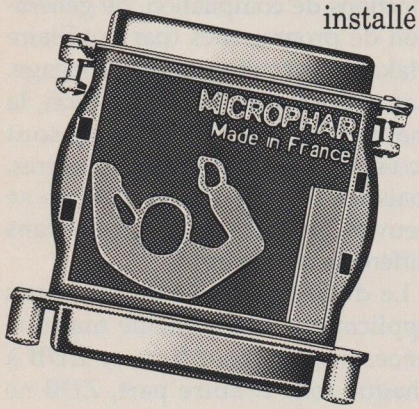
EXPÉRIENCE : Depuis 1983, 750 000 clés protègent les logiciels de 3 500 SSII et grandes entreprises.

SÉCURITÉ : Chacun de nos clients bénéficie d'une fabrication personnalisée, réalisée uniquement à sa demande.

ÉVOLUTION : Les routines d'interrogation de nos clés sont maintenues dans plus de 130 langages sous DOS, UNIX, OS/2 et WINDOWS.

PÉRENNITÉ : 23 personnes assurent le développement des produits et le support de nos clients.

INNOVATION : Nous venons de développer une clé permettant le contrôle du nombre d'utilisateurs d'un logiciel installé sur réseau.



- Clé Électronique et clé à Mémoire pour PC et PS,
- Clé pour Macintosh,
- Clé pour Réseau local,
- Clé à Microprocesseur à mémoire pour Stations de travail, Minis et tout ordinateur disposant d'un port série RS232.

MICROPHAR

n°1 européen de la protection des logiciels,
vend 55% de sa production dans 11 pays d'Europe et aux Etats Unis.

122, avenue Charles de Gaulle - 92200 Neuilly / Seine - Tél. : (1) 47 38 21 21 - Fax : (1) 46 24 76 91 - Belgique : 091 21 11 17 - Suisse : 024 21 53 86

Service lecteurs n° 27 (dernière page)

LANGAGE

ZORTECH C++ V3.0

Un compilateur multi-environnement

BERTRAND PIEDECOCQ

Zortech C++ supporte le développement d'applications pour DOS, Windows et OS/2. Il permet également de programmer en mode protégé grâce aux deux DOS Extender livrés avec le compilateur.

L'un génère des applications en mode protégé 16 bits (ZPM 286 DOS) et l'autre en 32 bits (DOSX). Ces deux modes brisent la frontière des 640 ko de mémoire imposée par MS-DOS.

Le compilateur est également compatible avec les deux DOS Extender les plus répandus sur le marché : le 386 DOS Extender de Phar Lap et le DOS 16 M de Rational Systems. Livré sous la forme de huit disquettes haute densité, l'installation du Zortech C++ s'effectue simplement grâce à un programme INSTALL. La mise en place de toutes les options nécessite plus de 20 méga-octets de libre sur le disque dur.

Un environnement moins performant que celui de Borland

Le temps de la ligne de commande pour lancer une compilation est désormais révolu. La plupart des compilateurs proposent des environnements pour l'édition des fichiers source et leur compilation, et certains vont même jusqu'à proposer des gestionnaires de projet évolués.

Symantec, grâce au Zortech C++, dispose maintenant d'outils de programmation multi-environnement pour les compatibles PC y compris en mode protégé.

Ces environnements sont plus ou moins sophistiqués et, dans ce domaine, celui du Turbo C++ de Borland fait office de référence : l'interface utilisateur, l'édition multi-fenêtre des fichiers source, la gestion des projets et le débogueur intégré simplifient, en effet, le développement d'applications complexes.

Les fichiers projet regroupent les fichiers source qui vont constituer l'application finale, les fichiers en cours d'édition ainsi que les bibliothèques à inclure au moment de l'édition de liens. Pour chaque fichier, des options sont accessibles pour indiquer quel compilateur employer et ses options de compilation.

Quant à l'environnement de programmation du Zortech C++, appelé «ZDB», il est pilotable par souris et comprend un éditeur multifenêtres avec les fonctions classiques Copier-Coller et Rechercher-Remplacer.

Le menu Compilation inclut des fonctions de compilation, de génération de programmes (par l'utilitaire Make), d'exécution et de débogage. Les options pour la compilation, la maintenance et l'exécution sont accessibles dans différentes fenêtres, mais les fichiers correspondants ne peuvent pas être sauvegardés dans différents répertoires.

Le développement de plusieurs applications sur la même machine nécessite de reconfigurer ZDB à chaque fois. D'autre part, ZDB ne dispose pas de gestion de projet. Pour la génération d'applications

Les compilateurs du C ++ ne sont pas l'apanage de Borland ni de Microsoft. Le développeur dispose, en effet, d'une troisième source par le truchement du Zortech C++ distribué par Symantec. Ce compilateur ne présente pas la même qualité d'ergonomie que ses compétiteurs. Mais il est accompagné d'utilitaires très performants pour l'optimisation du code et la conception de programmes s'exécutant en mode protégé. Ses bibliothèques fournies en standard couvrent de surcroît la plupart des besoins.

multifichiers, le développeur doit éditer son propre fichier Make, ce qui l'oblige à se référer à la documentation pour les options de compilation. Une fonction simple permettant de générer un fichier Make minimal selon les options de compilation et le fichier en cours aurait été très utile. A noter également l'impossibilité de définir des Define conditionnels lors de la compilation.

Le menu Browse propose, en revanche, trois fonctions intéressantes : le lancement d'un utilitaire Grep (recherche d'un texte dans un fichier texte), le C Tag Browse pour la recherche des définitions de classes, structures, etc., et enfin une option Browse Using qui affiche dans une fenêtre séparée les différents fichiers de définitions (*.H*) utilisés par le fichier en cours d'édition.

Pour en finir avec les désagréments, la boîte de dialogue d'ouverture d'un fichier n'autorise pas le changement de répertoire, et l'environnement n'est pas compatible avec les claviers non américains : il est très déroutant au départ de ne pas avoir accès au caractère «.». La découverte dans le menu Editor configuration d'une option Clavier non américain a résolu ce problème.

WINDOWS MAKER PROFESSIONAL

Une gamme complète d'outils de développement sous Windows est proposée par la société américaine Blue Sky Software. Le package Zortech C++ 3.0 comprend une version limitée d'un générateur d'applications sous Windows. Nous avons été très «frustrés» par cette version qui n'autorise pas la sauvegarde et la génération du code. Nous avons donc obtenu de la société Oser, qui diffuse en France ce produit, une version non limitée (V3.10).
Ces possibilités sont très intéressantes. Le programmeur trouvera tous les outils pour développer l'interface utilisateur de son application. Le code est ensuite généré, ou l'application créée tout simplement, par la sélection de l'option dans le menu «Fichier». Pour la génération d'applications, Windows Maker Professional reconnaît à l'installation les compilateurs C ou C++ de Microsoft, Zortech ou Borland. Suivant le compilateur, il sera capable de créer les fichiers Make et Link nécessaires à la réalisation d'une application. Le code produit est en C.
Une future version devrait générer du code C++, d'après ce qui nous a été affirmé. Plusieurs utilitaires sont livrés avec le package. Le premier QuickMenu sera familier aux utilisateurs de On Cue sur Macintosh sous système 6, du menu Pomme et des alias sous Système 7. Au lancement de Windows, un carré contenant le logo de Blue Sky Software se présente dans la fenêtre active. Un clic sur celui-ci fait apparaître un menu configurable permettant le lancement des applications Windows ou non disponibles. Il sera difficile de s'en passer après la première utilisation, et cet utilitaire mériterait même une diffusion plus large!
Windows Maker Professional Editeur : Blue Sky Software,
distributeur exclusif : Oser - Paris 10e, prix (HT) : 8 900 francs.

Le compilateur et ses programmes associés

Le compilateur (ZTC) est accessible par la ligne de commande et se révèle plutôt rapide, mais la vitesse de compilation et l'édition de liens demeurent légèrement inférieures à celles de Turbo C++. A noter qu'une option de compilation (variable d'environnement SET TEMP=) permet d'utiliser un éventuel disque virtuel pour le stockage des fichiers temporaires. ZTC est compatible avec la norme Ansi et génère des fichiers objets au standard Microsoft ou Phar Lap (32 bits).

Les instructions de débogage pour CodeView, le débogueur de Microsoft, peuvent être incluses lors de l'édition de liens. Pour les très grosses applications sous DOS, des fichiers de recouvrements de code (overlays) peuvent être créés. Chez Zortech, ce dispositif s'appelle Virtual Code Manager et donne une alternative aux DOS extenders, mais avec des limites : 255 segments virtuels de code sont accessibles, chacun ayant une taille maximale de 64 ko.

Dans le domaine des options de compilation, il est possible de géné-

rer des instructions spécifiques aux processeurs 286, 386 et 486, ainsi qu'un optimiseur de code. Celui-ci est un programme indépendant, et son utilisation revient à ajouter une passe dans la compilation. Il donne le choix entre un code plus compact ou une meilleure vitesse d'exécution.

En raison de ses fonctions très évoluées, l'optimiseur se révèle lent d'exécution. En particulier, il est capable de supprimer totalement les lignes inutiles dans un programme ou une fonction (variables locales non utilisées, etc.).

Pour les opérations en virgule flottante, le compilateur est livré avec une bibliothèque d'émulation d'un coprocesseur mathématique. Une option de compilation permet également d'inclure des instructions 80 x 87 directement dans le code. Dans ce cas, la bibliothèque est inutilisée.

MAGIC FIELDS ET ROBOHELP

La société Oser diffuse un add-on à Windows Maker Professional : Magic Fields et roboHELP qui est un générateur de fichiers d'aide Windows. Magic Fields comprend un gestionnaire de contrôles évolués, en particulier des champs de saisie suivant des formats prédéterminés tels les dates, heures, numériques (décimaux ou hexadécimaux), ... ou des formats définis par le développeur.
Celui-ci dispose aussi des fonctions de définition des modes d'édition, des bornes de valeurs minimales et maximales et des masques de validation des données.
Cet add-on est livré sous la forme d'une extension à l'éditeur de dialogues et la rubrique Magic Fields apparaît alors dans la liste des contrôles.
La définition d'un contrôle Magic Fields provoque l'apparition d'une boîte de dialogue de définition du contrôle et de l'environnement. Notons qu'il est possible de donner un «look» NeXT aux contrôles, voire à la boîte de dialogue entière. RoboHELP propose des outils pour la mise en œuvre facile des écrans d'aide hypertexte classiques à Windows. Un modèle de document sous Word est fourni. A l'ouverture de celui-ci, une palette d'outils est affichée et sert à créer une table des matières, une table des définitions ainsi que les liens hypertexte. Finalement, RoboHELP se charge de la création du fichier d'aide au format RTF, du fichier de déclaration des constantes et d'un fichier de projet pour le compilateur d'aide.
Magic Fields Editeur : Blue Sky Software, distributeur : Oser - Paris 10e
prix (HT) : 2 800 francs.
RoboHELP Editeur : Blue Sky Software, distributeur : Oser - Paris 10e
prix (HT) : 3 800 francs.

Les outils de développement

Deux versions du débogueur (ZDB) sont incluses, l'une strictement DOS et l'autre utilisant le DOS Extender dans sa version 286. Le débogage de grosses applications se fait plus facilement avec la deuxième version, sous réserve que la machine utilisée soit employée de mémoire étendue.

Au lancement de ZDB, celui-ci crée ses propres fichiers contenant les listes des différents labels de variables et fonctions. Cette opération est relativement longue et plutôt fastidieuse puisque répétée après chaque modification.

LES SUBTILITÉS DE PROGRAMMATION EN MODE PROTÉGÉ

Des outils de développement existent maintenant afin de générer des applications tournant en mode protégé et utilisant la pleine puissance des processeurs de la gamme 80 x 86 à partir du 286. Le mode protégé détruit la fameuse limite des 640 ko et l'application peut accéder à toute la mémoire vive installée dans la machine : l'adressage des données s'effectue non plus en 20 bits mais en 24, voire en 32 bits. Les processeurs 286 disposent d'un mode protégé 16 bits alors que les processeurs 386 et 486 intègrent les modes du 286 ainsi qu'un mode protégé en 32 bits. Toutes les fonctions du DOS dont les arguments sont des pointeurs d'adresses sont réécrites afin que ces dernières soient compatibles avec le nouveau mode d'adressage en mode protégé. Par exemple, dans la fonction 3Fh de l'interruption 21h (lecture dans un fichier ou un device), le buffer en mode réel est pointé par les registres DS:BX. Le buffer de la nouvelle fonction est pointé par les registres DS:EBX et le nombre d'octets à lire est stocké dans le registre 32 bits ECX au lieu de CX. La gestion des interruptions et celle des device drivers écrits en mode réel sont pris en charge par le nouveau système, le DOS Extender se configurant en mode réel avant de passer la main aux drivers.

L'interface utilisateur est de type multifenêtre et reconnaît la souris. Différentes fenêtres permettent de visualiser les données, l'état du processeur ou du coprocesseur, de la mémoire et des variables. La fenêtre Variables a un comportement assez particulier puisque la liste de toutes les variables globales est affichée avec leurs adresses. Pour en visualiser le contenu, il faut frapper la touche «+». On accède alors au contenu des variables ou structures. La touche «-» permet de supprimer des variables ou une portion de structure afin de simplifier cette liste.

Une fenêtre Write Protect visualise les différentes zones mémoire protégées contre l'écriture, et la fenêtre Expressions visualise, pour sa part, des expressions qui peuvent être une opération entre deux labels ou variables globales. Quant aux points d'arrêt, ils sont soit sur une simple ligne de code, soit conditionnels. Là aussi, une fenêtre affiche les conditions d'arrêt. A la

Plus de 1500 produits de développement présentés dans le catalogue le plus complet du marché

TECHNO-DIRECT

VOTRE MEILLEUR FOURNISSEUR DE LOGICIELS DE DEVELOPPEMENT

- *La garantie des prix les plus bas : nous nous alignons sur tout prix du concurrent publié le même mois.*
- *Un stock important pour vous livrer rapidement.*
- *Des spécialistes prêts à vous écouter et à vous conseiller.*
- *Un service "TECHNO-FAX" inédit vous permet d'obtenir gratuitement à partir de votre télécopieur toute documentation sur le logiciel de votre choix au (1) 45. 06.0 67. 01.*
- *Un service "FOUINEUR" gratuit vous confirme en moins de 48 heures, prix et délais de livraison de n'importe quel logiciel ou matériel que vous recherchez aux USA.*

dBASE/FOX/CLIPPER

	Prix TD TTC.
CLIPPER 5.1, VF	9 120
dBASE IV + Ashton tate, VF*	7 531
FOXBASE 2.1	3 546
FOXPRO, Fox Software, VF, 2.0 PROMOTION*	7 450
R&R, VA	1 767
CLEAR FOR dBASE, Clear S, VA	1 767

C/C++

MICROSOFT C/C++ 7.0, NOUV. VERS.	2 953
MICROSOFT QUICK C	818
TURBO C++, 2nd édition, Borland, VF	943
QUICK C Windows, Microsoft, VF	1 648
C++ développement, ZORTECH, VA	4 530

OUTILS C++

HIGH SCREEN 5, PC SOFT, VF	4 685
METAWINDOWS T/C, Metagraphics	1 174
CODE BASE 4, INNOSOFT, VF	2 241
C ASYNCH MANAGER, Blaise, VA	1 648
C TOOLS PLUS, Blaise, VA	1 352
HYPERPRINT 2, PC SOFT, VF	4 684
CODE BASE C++, VA — Une bibliothèque ISAM compatible avec les index NDX et MDX de dBASE III et IV, compatible C++ et Windows	2 241
WIN++, Blaise, VA — 100 classes objets pour gagner en productivité sous C++	2 241
TOOLS L++, VA — 60 classes pour développer plus vite en C++, DDE, compatibilité Windows 3.0.	1 055
RT LINK Plus, Pocket Soft, VA, PROMO*	3 107

EDITEURS

SPF/PC, Command Technology, VA	2 550
Sage Professional Editor, VA (DOS)	2 122
BRIEF 3.0, VF, NOUVEAU	1 767
BRIEF, 3.1, US	1 886
dBRIEF, US (dBASE ou Paradox)	1 061
BRIEF C++, US, nécessite BRIEF	1 061

PLUS DE 3000 PRODUITS A NOTRE CATALOGUE

BASIC

QUICK BASIC 4.5, Microsoft, VF, PROMO*	818
POWER BASIC 2.0, VF	1 174
MICROSOFT BASIC, PDS, 7.1, VF	3 309
DBLIB, Ajs, VA	1 174
TURBO SCREEN, PC Soft, VF	1 174

FORTRAN

FORTRAN 5.1 Microsoft, VA, nouv. vers	3 368
LAHEY F77L-EM 32	12 987
PRINTMATIC, Micrompatibles, VA	1 482
MATHLIB, Wiley, VA	2 360

L-LINKERS / DOS EXTENDERS

386 DOS EXTENDER KIT	4 950
BLINKER, VA	2 241
RT LINK/Plus 4.10, VA	3 783

PASCAL

	Prix TD TTC.
PASCAL COMPILER, Microsoft, VA	3 309
QUICK PASCAL, Microsoft, VF	818
TURBO PASCAL V6, Borland, VF	1 150
TURBO PASCAL PRO V6, Borland, VF	2 307
TURBO PASCAL Windows, Borland, VF	2 004
DATABOSS 3.5, Innosoft, VF	6 390
SCIENCE & Engineering Tools, VA Quinn Curtis	1 180
TURBO BTREE, VF — Vous permet d'écrire vos applications de base, deux milliards d'enregistrements, 100 clés par fichier index, support réseau.	1 886
TURBOVISION DEVELOPPEMENT KIT — Blaise. Boîte à outils, complément idéal de Turbovision : boîte de dialogues, menus, intégration d'objets, possibilité de modification sans recompilation. Source inclus. Pas de royaltie.	1 411

PRODUITS SCIENTIFIQUES/CAO

AUTOCAD 11, VF	33 445
ANIMATOR	2 846
DESIGN CAD 2D, VF	3 783
Cross Assembler 2500 AD, VA	2 190 à 4 590
GRAPHER, Golden Software, VA	2 004
GRAPH IN THE BOX, ANALYTIC, VA	1 886
MATHCAD VA , Traitement de texte	4 732
STATGRAPHICS, STSC,	9 369
SPSS, Base 4.0, VA	2 957
TK SOLVER PLUS, UTS, VA	4 376
PSN 5, Scitor, VF	9 002
MATHEMATICA 387, Wolfram, VA	11 730
MATHWORD, Apsylog, VF	3 309

TOUTE LA GAMME PC SOFT

HIGH SCREEN 5.5 — Générateur d'écrans	4 685
HYPERFILE 2.0, SGBD	4 685
HYPERPRINT 2.0 — Générateur d'états	4 685
HYPER PACK Développeur — Atelier génie logiciel : High Screen + Hyper File + Hyperprint	9 369

WINDOWS APPLICATIONS

BECKERTOOLS WINDOWS	818
WINDOWS EXPRESS, HDC, VF	937
CROSSTALK WINDOWS, VA	1 649
TWINTALK, VF	2 360
DESIGNER, VF	7 590
WINDOWS DRAW, VF	2 597
SUPER BASE 4, VF	6 630
WORD WINDOWS, VF	3 677
LOTUS WINDOWS, VF	3 997
MULTIMEDIA, dév. kit , Microsoft	4 139
ADOBE TYPE MANAGER	747

PLUS DE 2000 PRODUITS EN STOCK

UNIX/XENIX

	Prix TD TTC.
MKS TOOLKIT, MKS, VA	2 718
XENIX 386, SCO, PROMO*. Dévelop.	6 084
386/IX-TCP/IP, Interactive, VA	4 270
UNIX SYSTEM V, SCO, PROMO* Runtime, 2 ut, VA	4 969
SCO TCP/IP, Développement, PROMO*	1 993
SCO X WINDOWS, PROMO* Développement	3 114

WINDOWS - OUTILS

ACTOR 4.0, VA	2 360
CASEWORKS, VA	5 870
COMMONVIEW C++, VA	5 918
DROVER' TOOLBOX, VF	3 190
HIGH SCREEN 5/WINDOWS, VF	8 052
MULTISCOPE WINDOWS, VA	3 190
TOLLBOOK, VF	4 020
VISUAL BASIC, VF	1 530
VISUAL BASIC SQL SERVER	3 427
VISUAL JT BASE,	5 455
WINDOWS SOFT DEV. KIT	3 143
WINDOWS DEVICE KIT	3 143
WIN++/Blaise	2 241

TOUTES LES MISES A JOUR DISPONIBLES

UTILITAIRES

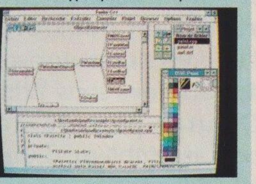
IMPRIM'DOC, Innosoft, VF	943
386 MAX PRO (VF : ☎)	747
COPYWRITE, Quaid, VA	925
PC TOOLS 7.1, VF, NOUVEAU	1 530
DATABRIDGE PLUS, VF, PROMO*	1 992
NORTON ADV 6.0, VF	1 293
ADOBE TYPE MANAGER, PROMO*	735
PIZZAZ PLUS, VF	1 886
LAPLINK PRO, VF, nouvelle version	1 293
TECHTAB, PROMO*	3 469
SPINWRITE II	854
MENUMAKER +, VF getionnaire de menus	943
MACPRINT, Insight, VF	1 376
FASTBACK PLUS, VF (3.0, VA : ☎)	1 174
NORTON COMMANDER, 3.0, VF	937
DR DOS, 6.0	695
QEMM 386, VF	1 056
DOS 5.0	592

DIAGNOSTICS NOUVEAU

ALIGN IT	1 648
KICKSTART 1	699
PC PROBE	1 174
WINSLEUTH PRO	1 530

TURBO C++ WINDOWS

Créer des programmes Windows devient rapide et simple grâce à Turbo C++ pour Windows. Des outils puissants et intuitifs du type "pointez et cliquez" rendent la programmation Windows plus simple et accessible même si vous n'avez jamais programmé avant : environnement EDI intégré, Object windows, Object Browser, Easywin et Resource Workshop.



- Turbo C++/W : **1 095 F** HT
- Mise à jour : **895 F** HT

BORLAND C++ 3.0

Nouveau compilateur C++ avec support complet du CANSI et du C++ 2.1, un optimiseur global puissant, les templates et de meilleures performances, Borland C++, est l'un des systèmes de création de code les plus productifs, qu'il s'agisse d'un environnement DOS ou Windows. Turbo Debugger, Turbo Profiler, Turbo Assembler inclus.

- C++ 3.0 : 3995 **2 795 F** HT
- Mise à jour en C++ 3.0 : **995 F** HT
- Borland C++ 3.0 + Application Framework + cassette C++ : **4 190** F HT

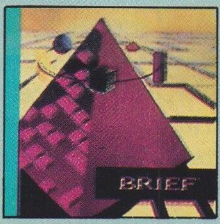
OBJECT PROFESSIONAL PASCAL C ou C++

Object Professional inclut plus de 100 types d'objets qui permettront d'augmenter votre productivité : interfaces de type PM ou Windows, support souris, menus déroulants, générateurs d'écrans de saisie et de menus, gestion de fichiers d'objets, impression de formulaires, sous-programmes pour créer des résidents avec swapping dans 6 Ko. Object Professional existe pour Turbo Pascal ou C++. Code source inclus. Pas de royalties. Support technique gratuit.

- Version Pascal, VF : **2 095 F** HT
- Version C++, US : **1 995 F** HT

BRIEF FRANÇAIS ou US

BRIEF est le seul éditeur de texte orienté programmation du marché à offrir aux développeurs un macrolangage complet adaptable à pas moins de 52000 styles différents de programmation !
Son langage macro est en effet unique et facile à utiliser.



- Brief 3.0 VF : **1 490 F** HT
- Brief 3.1 US : **1 590 F** HT
- Mise à jour Brief : **1 190 F** HT

PVCS

PVCS est le meilleur gestionnaire de projets sur le marché. Il permet le stockage et la gestion de multiples version d'un fichier source. Il tient à jour un historique des modifications, et peut éditer des rapports.
PVCS existe en version réseau.

- PVCS DOS : **3 600 F** HT

TVA 18,6%

NOUVEAUX CATALOGUES

CAT. 11

- ☐ Je désire recevoir votre nouveau catalogue PC N° 11
- ☐ Je désire recevoir votre nouveau catalogue MAC N° 4

1Sf7

Nom..

Société..

Fonction..

Adresse..

....

CP.................... Ville..

Tél. ....

CONDITIONS D'ACHAT : Horaires d'ouverture : 9h-19h du lundi au vendredi. CB acceptées contre R avec supplément. Possibilité d'enlèvement sur place : Immeuble Eiffel - 6, bd Henri Sellier - 92 150 Suresnes. Prix donnés pour départ Suresnes, paiement comptant. Frais de port France métropolitaine 60 F (71,16TTC) pour toute commande inférieure à 1 500 F, 160 F HT (177,90 F TTC) pour toute commande comprise entre 1500 F et 5 000 F. Nous téléphoner pour les commandes supérieures et pour toute expédition à l'étranger.Le matériel livré est assuré par nos soins. Prix indicatifs modifiables sans préavis. Nous nous alignons sur tous prix fermes donnés par des conccurents.
Tous les prix sont indicatifs et modifiables à tout moment - (VF : %) : appelez nous pour les prix des versions françaises
***Les promotions sont limitées dans le temps et dépendent du stock disponible. Nous consulter**

mc2 - 49.10.09.46

TECHNO-DIRECT - Téléphone : (1) 40 99 28 28 - Fax : (1) 40 99 28 88

Service lecteurs n° 28 (dernière page)

longue, cette multiplication de fenêtres finit par être fatigante.

Notons, enfin, qu'une option permet de déboguer à distance un logiciel par l'intermédiaire d'un câble série reliant deux ordinateurs et qu'une version autorise le debogage d'applications sous Windows.

D'autre part, le débogage en mode assembleur d'un programme écrit en langage C ne commence pas automatiquement à la fonction main(), et le débogage de programmes écrits pour le Phar Lap 386 Extender n'est pas possible, ce qui est très dommage : Phar Lap ne livre qu'un débogueur symbolique équivalent au SYMDEB de Microsoft dont l'emploi n'est pas vraiment simple.

Les librairies de procédures

Le Zortech C++ comprend une bibliothèque de fonctions pour le développement d'applications graphiques : Flash Graphic. La liste des modes et cartes graphiques est plus à jour que celle des drivers BGI du Turbo C++ : elle reconnaît les modes VESA, les cartes 8514 ou TIGA, les résolutions étendues Paradise et Orchid, etc. Elle intègre également des fonctions plus évoluées telles que la gestion de la souris en mode graphique. Contrairement aux programmes compilés en Turbo C++ et qui demandent la présence des fichiers .BGI, les applications écrites en Zortech C++ comprennent la bibliothèque graphique dans le code. Cependant, la procédure d'autodétection matérielle n'a pas fonctionné correctement avec deux machines différentes (un ALR 486 équipé d'une carte SuperVGA à base d'un circuit ET4000 et un PC compatible taiwanais). Les appels aux fonctions simples comme le tracé des lignes comportent trop de paramètres : la couleur, le mode et le masque de tracé doivent être passés à chaque appel de la routine, ce qui ne simplifie pas la programmation et augmente la taille du code généré.

Des fonctions annexes permettant de changer les paramètres auraient été préférables. D'autres bibliothèques donnant accès à des fonctions d'affichage, de gestion des fichiers, des accès au DOS ou au Bios, etc. sont livrées avec le Zortech C++. Ces fonctions sont réunies dans plusieurs groupes. On y trouve notamment le «Display Package» qui gère l'affichage en mode texte, l'«Emm Package» qui pilote les accès à la mémoire paginée (EMS 4.0) et l'«Handle Package» qui intègre des fonctions recopiant dans la mémoire EMS des zones mémoires. Regrettons que la mémoire XMS ne soit pas utilisée, car elle s'avère plus rapide. Par ailleurs, en l'absence de mémoire EMS, un swap sur disque dur n'est pas possible.

Citons, enfin, un gestionnaire d'interruptions, de souris et des fonctions pour l'installation de programmes résidents (TSR). Ces groupes ne sont pas encapsulés dans des classes. Pour le «véritable» développement orienté objet, plusieurs classes offrent des fonctions de gestion des flux, des nombres complexes, etc. Enfin, des classes baptisées «Tools C++», livrées sous forme de fichiers source, comprennent la définition et le code de classes facilitant la conception d'applications C++. On y trouve ainsi des classes de gestion de champs de bits, d'entiers BCD, de listes, d'arbres B Tree, de variables virtuelles (stockées sur disque et basées sur le mode Dernier objet lu), de fenêtres texte, de menus et

LES DOS EXTENDER DE PHAR LAP ET DE RATIONAL SYSTEMS

La société Phar Lap commercialise depuis longtemps deux DOS Extender : une version 286 et une version 386. La version de base du 386 DOS Extender comprend un assembleur, un éditeur de liens, un débogueur symbolique inspiré du SYMDEB de Microsoft et d'autres utilitaires. Ce programme est compatible avec les différents standards du marché : DPMI (Windows en mode étendu), VCPI (gestion de mémoire étendue), XMS, et il supporte l'INT 15h pour l'allocation de mémoire étendue. Les programmes sont «linkés» dans un fichier dont l'extension est .EXP et se trouvent lancés par la commande RUN386 suivie du nom du programme. Le logiciel RUN386 effectue les initialisations, commute le processeur en mode protégé, charge et exécute le programme désiré. Une version «run time» existe et permet la création d'un fichier exécutable, mais la diffusion de l'application reste limitée à 1 000 exemplaires.
Une diffusion plus importante est à négocier avec le fabricant. De nombreux compilateurs sont maintenant compatibles avec ce système, en particulier le Zortech C++ et la gamme des compilateurs de Metaware (pascal, C, fortran,...).
La société Rational Systems diffuse, de son côté, un DOS Extender pour les processeurs 286 et supérieurs. Il permet de développer des applications en mode protégé, adressant jusqu'à 16 méga-octets de mémoire. Les adresses en mode réel sont converties pendant l'édition de liens en adresses en mode protégé. Les appels aux fonctions DOS sont maintenus, le programme commutant le processeur en mode réel avant chaque appel à une fonction DOS. Les logiciels générés avec ces outils présentent donc l'avantage d'accéder à une plus grande plateforme de machines, mais on peut craindre que l'exécution sera ralentie par ces commutations.
Phar Lap 386 DOS Extender : 5 590 F HT (run time : 21 950 F HT)
286 DOS Extender : 5 590 F HT, distribué par Micro-Sigma, Techno Direct.
Rational Systems DOS LIB 16M : 50 000 F HT (200 exemplaires), distribué par Micro-Sigma.

de champs. Les bibliothèques de Zortech C++ sont très évoluées, mais longues à assimiler.

Le développement sous Windows

Le package Zortech C++ ne comprend pas à proprement parler d'outils de développement spécifiques pour Windows. En fait, le programme d'installation recherche la présence des utilitaires du SDK.

Des applicatifs Windows lancent les compilateur, débogueur, et work-bench sous DOS après avoir demandé, dans une boîte de dialogue, les paramètres de la ligne de commande. Tous les outils sont présents comme le compilateur de ressources ZRC, le compilateur d'aide Microsoft HC et un volume reproduisant le Microsoft Windows Function Reference.

Cependant, l'absence d'outils d'aide tels qu'un éditeur de ressources rend obligatoire la présence du SDK ou de toute autre aide au développement. Si l'on se base sur les outils de développement sous Macintosh Think C et Think Pascal de Symantec, on peut envisager des améliorations dans les futures versions du Zortech. Une bibliothèque de classes équivalente pour le développement d'applications sous Windows pourrait être incluse. Celles-ci encapsulent tous les gestionnaires disponibles dans la boîte à outils du Mac et facilitent grandement le développement des applications.

On pourrait donc imaginer la même chose sous Windows et, pourquoi pas, des classes communes autorisant un portage simplifié d'un environnement à l'autre.

En conclusion, grâce aux différents environnements supportés, le Zortech C++ peut s'avérer un bon choix. Regrettons simplement qu'il manque parfois de simplicité et qu'un puissant gestionnaire de projet ne s'y trouve pas inclus. ❑

TopSpeed

Le spécialiste de la compilation

Générateur optimisé Haute qualité du code généré, optimisation de l'édition de liens (toute variable ou fonction non utilisée n'est pas incluse dans l'exécutable).

Plus de limite à 640 Ko Selon que vous choisissez de travailler en mode réel ou mode protégé, la mémoire disponible pour vos programmes peut atteindre de 16 MO à 512 MO. TopSpeed gère automatiquement les overlays pour le code et les données, les librairies dynamiques (DLL) chargées et déchargées à la demande et utilise même le disque en mémoire virtuelle de vos programmes.

Multi-tâches Jusqu'à 32 tâches concurrentes même sous DOS.

Code source disponible Toutes les librairies peuvent être fournies en code source.

Portabilité Génère du code pour DOS, OS/2, et Windows 3.

Multi langages Choisissez le langage le mieux adapté au traitement que vous allez réaliser; TopSpeed permet de construire une application à partir de modules écrits dans des langages differents (C, C++, Pascal, Modula-2). Utiliser n'importe quelle librairie à partir de n'importe quel langage. Développez dans des langages différents en utilisant le même éditeur et le même environnement.

Environnement *TopSpeed* Commun à tous les langages, éditeur 10 fenêtres d'un Mo chacune, gestionnaire de projet, aide hypertexte (langage et librairie), debugger, contrôle de syntaxe, TSR pour calculatrice, table Ascii, scan-code, etc.
DOS ou OS/2 **590,00 F**

***TopSpeed*C** Certifié ANSI pour portabilité maximum, génération automatique des prototypes, compatible Turbo C, Microsoft C 5.1 et Quick C, génère du code compact de haute qualité pour DOS, OS/2 et Windows 3, contrôle des anomalies à l'exécution (pointeur nul, dépassement de bornes, etc).
DOS ou OS/2 **590,00 F**

***TopSpeed* C++** Conforme AT&T 2.1, programmation orientée objet. Editeur de liens **Smart Method** élimine automatiquement les classes et méthodes non référencées dans le code.
DOS ou OS/2 **590,00 F**

***TopSpeed* Pascal** Conforme ISO 7185 Niveau 1, extensions orientés objets, array et string dynamiques, convertisseur Turbo Pascal, compilation séparée des unités, etc.
DOS ou OS/2 **590,00 F**

***TopSpeed* Modula 2** Successeur de Pascal, langage très structuré et orienté objets, conversion de type automatique.
DOS ou OS/2 **590,00 F**

TopSpeed* TechKit** DLL même sous DOS, Debugger Post-Mortem (permet d'analyser les causes d'un «plantage», Profiler (permet d'optimiser la vitesse d'exécution de votre programme en détectant les sections les plus coûteuses en temps), assembleur ***TopSpeed, compresseur d'EXE, désassembleur, librairies et compilateur de ressources pour Windows 3.
DOS ou OS/2 **590,00 F**

Nouveau

***TopSpeed* DOS Extender**

Fonctionnement en mode protégé sur processeur 80286 et **au delà**. Jusqu'à 512 MO de programmes et données chargés et déchargés automatiquement à la demande de l'application. Supporte XMS, DPMI, VCPI, API OS/2.
DOS ou OS/2 **1180,00 F**

Offre Promotionnelle

☐ Mode Réel		☐ Mode Protégé	
Environnement	590,00	Environnement	590,00
+ 1 compilateur	590,00	+ 1 compilateur	590,00
		+ DOS Extender	1180,00
	~~1180,00 HT~~		~~2360,00 HT~~
	990,00 HT		1990,00 HT
+ TVA + frais d'envoi	**1249,14 TTC**		**2435,14 TTC**

Bon de commande

Veuillez trouver ci-joint mon règlement par chèque pour l'offre promotionnelle de ☐ MODE Réel ☐ MODE Protégé – en version anglaise à l'ordre de :

BIONIX – 32, rue Molitor – 75016 Paris
Tél : 40 71 09 36 Fax : 46 51 24 26

TopSpeed® C++ TopSpeed® Modula-2 TopSpeed® Pascal TopSpeed® C

Service lecteurs n° 29 (dernière page)

I & s 7

GOLD 92

LE GRAND LOGICIEL ET

Le principe de l'«Open» est de permettre aux développeurs, lecteurs de *Langages et Systèmes*, de présenter une application professionnelle qu'ils ont réalisée en vue de l'obtention du GOLD 92 de *Langages et Systèmes*.

LES MODALITÉS DU GOLD 92

Il comporte deux pools. Chacun d'eux se verra attribuer un GOLD 92, plus deux autres prix.

1er pool : les logiciels développés au moyen d'un langage figurant dans la liste ci-dessous :

Assembleur
C (versions DOS ou Windows)
C++ (versions DOS ou Windows)
Pascal (versions DOS ou Windows)
QuickBasic
Visual Basic

2e pool : les logiciels développés au moyen de générateurs d'applications commercialisés sous la forme de package, de type Windows Toolbook, NewWave, Clipper, dBase, Foxpro, Paradox, Oracle, Magic II, Servant IV, High Screen, etc. Dans ce pool, l'utilisation d'un langage interfacé ne devra pas excéder 20% du code total de l'application développée.
Les candidats pourront — à l'aide de ces outils — présenter des applications indifféremment sous DOS, Windows et OS/2.

OPEN DÉVELOPPEMENT

RÈGLEMENT DU GOLD

Date limite de dépôt
Envoi de votre dossier d'inscription accompagné de votre logiciel à Langages et Systèmes.
Date limite : 31 août 1992.

Dossier
Votre dossier doit comporter les pièces suivantes :
1- La fiche d'inscription
Dans le cadre d'une participation en équipe, l'inscription doit se faire au nom du responsable du projet. Une fiche d'inscription sera remplie pour chaque logiciel présenté.
2- La fiche d'identité du logiciel
Une fiche d'identité doit être remplie pour tout logiciel présenté.
3- Deux exemplaires de votre logiciel sur disquette (de préférence 3,25 pouces)
4- Les sources du programme sur disquette
5- Une version de démonstration de votre logiciel
Celle-ci sera présentée sur notre serveur 36 16 INFOPC.

Pour obtenir votre dossier d'inscription, adressez votre demande à :
GOLD / Langages et Systèmes
Cedex 65-92051 Paris la Défense

Les domaines d'application concernés
Les applications développées pourront concerner les domaines suivants : gestion, scientifique, enseignement assisté, bureautique et utilitaires systèmes.

Les critères de sélection
- Intérêt pour les utilisateurs - Originalité de l'application - Convivialité de l'application - Documentation et aide en ligne - Ergonomie de l'interface - Intégrité de l'application - Exploitation des ressources matérielles.

Le jury
Le jury du GOLD sera constitué des personnes suivantes :
- Le rédacteur en chef de Langages et Systèmes
- Notre partenaire éditeur Microsoft
- Un représentant de chacun de nos partenaires du concours : Victor, Commodore et Borland.
Les décisions prises par le jury seront sans appel.

Les prix
Dans chacun des pools définis, le vainqueur se verra décerné le GOLD 92 de Langages et Systèmes.
1er prix : Victor Technologies offrira une station multimédia pour le vainqueur du premier pool. Commodore offrira un PC 486 au vainqueur du second pool. PC Soft offrira des packages de développement DOS et Windows. Microsoft et Borland offriront des packages de développement.
2e et 3e prix : Microsoft et Borland offriront des packages de développement. PC Soft offrira High Screen DOS et Windows.
La remise des prix se fera au cours du salon Software Development qui aura lieu du 15 au 17 octobre 1992.

Langages et Systèmes *se réserve tout droit d'annulation du Grand Open Logiciel et Développement et dégage toute responsabilité pour la non-remise des prix mentionnés ci-dessus.*

INFO'PC

LE MAGAZINE DE LA MICRO-INFORMATIQUE PROFESSIONNELLE

Langages et Systèmes
Immeuble La Fayette 2, place des Vosges, cedex 65, 92051 Paris-la Défense 5
Tél : 49 04 79 00 - Télex : 613 234 F MONDINF

Directeur : Jacques TILLIER assisté de Catherine BOISADAM
Rédacteur en chef : François Yves LE GAL, **Assistant de la rédaction** : Serge DEL'HOMMEAU
1re secrétaire de rédaction : Catherine OLZCHANECH
Secrétaires de rédaction : Elisabeth DEROFF-VASSEUR, Isabelle DUMONTEIL, Stéphanie GRIN
Assistant de la fabrication : Stéphane GAUTIER
Ce numéro a été réalisé sous la responsabilité de Yves Guittard
Ont également collaboré à ce numéro
Marc Adler, Thierry Bienfait (rédacteur-réviseur), François Bergeon, Gilles Clavel, Maxime Daddler, Jean-François Duvivier, Eric de Kervalec, Christophe Lepecq, Bertrand Piedecocq, Henry-Luc Planchat, Arnaud de la Pommeraye, Norbert Rimoux, Matthieu Thfoin

La rédaction d'INFOPC dégage sa responsabilité en ce qui concerne les documents écrits, les matériels et les logiciels qui lui seraient envoyés sans demande préalable de sa part.

PUBLICITÉ
Directeur de la publicité : Sophie TALAYRACH, **Directeur adjoint de la publicité** : Nathalie TESSLER
Chefs de publicité : Virginie BRIL, Thien GILBERT, Hervé LABATTUT
Assistantes de publicité : Aromi RAJON, Laurence CARTON

DIFFUSION
Directeur : Jean PAITEL; **Responsable service abonnements** : Lucienne BOSSER assistée de Josianne LIBAN, Rose MOLNAR, Alain PACCAGNELLA, Bernadette THUAL; **Responsable des ventes** : Pierre-Olivier PROVENT

Abonnements : 49 04 79 00, 49 04 79 00; Suisse : MADITEC SA; Case postale-144CH 1020 RENENS VD1-; Tél : (021) 49 04 79 00

InfoPC est une publication éditée parMicropresse S.A. au capital de 250 000 F, en location-gérance auprès de

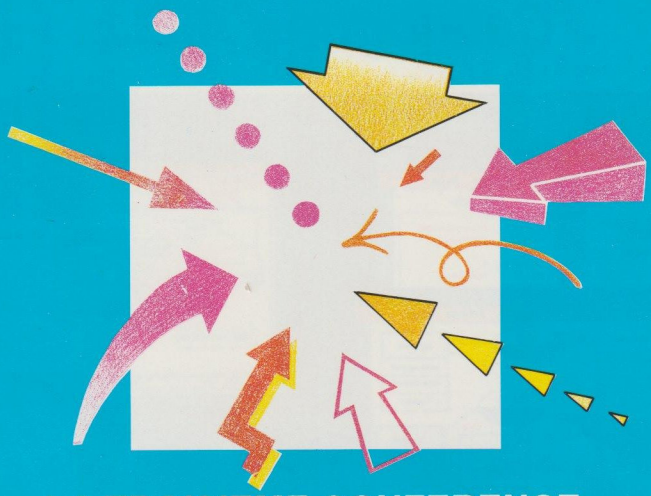
IDG COMMUNICATIONS FRANCE

ISSN 0981 - 6402
Commission paritaire n° 67338

INDEX PUBLICITAIRE

ALADDIN	36
ALSYS	8
ALTISOFT	47
BIONIX	107
CAP SESA	13
EDITIONS WEKA	99
INNOSOFT	64
MICRO SIGMA	11, 52, 75
MICROPHAR	101
MICROSOFT	2, 18
NOVELL	69
NSL	28
OSER	27
P7I DEVELOP.	87
PB NET	15
PC SOFT	6
PC TECHNOLOGIE	73, 111
RANK XEROX	112
REM CORP	92
SMARTLINE	12
SOFTWAY	52 à 60
TAU CETI	4
TECHNO DIRECT	105
TECHNOTRUFFE	35

PC TECHNOLOGIE, éditeur exclusif de Nantucket en France,
organise votre premier contact avec ASPEN / Clipper-Windows
et vous présente les nouvelles évolutions de Clipper 5



LA TROISIEME CONFERENCE ANNUELLE DES DEVELOPPEURS NANTUCKET-CLIPPER

12 / 13 / 14 mai 1992
Château de Montvillargenne
CHANTILLY

Pour tout complément d'information contacter

PC TECHNOLOGIE
4,4bis rue Bourgeoise 92800 PUTEAUX
Tel 42 04 00 88

PLURIELS ORGANISATION
19 rue Godefroy 92800 PUTEAUX
Tel 47 74 80 56

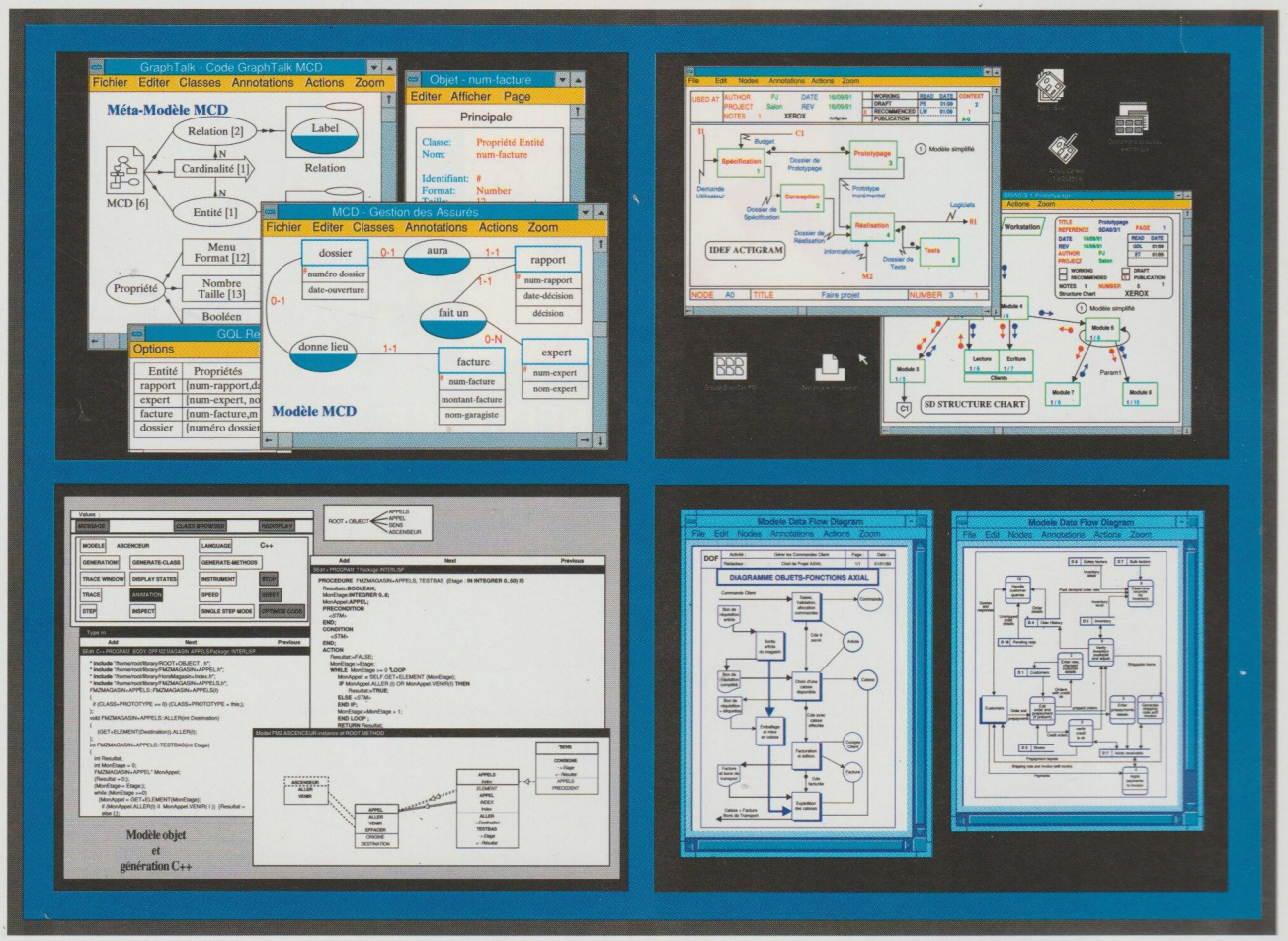
nantucket

Service lecteurs n°12 (dernière page)

RANK XEROX

La technologie GraphTalk

Générateur d'ateliers
Générateur de codes
Générateur de documents
Catalogue d'A.G.L.



Ateliers GraphTalk:
MERISE, AXIAL, NIAM, SA, SSADM, IDEF, SD, ROOT, OSSAD.

Plates-formes supportées:
PC, PS, stations de travail supportant OS2/PM, UNIX/MOTIF, MS/Windows.

Compatibilité:
Tout développement réalisé sur un système est immédiatement opérationnel sur les autres sans conversion.

Rank Xerox DIA/GL 7, rue Touzet 93586 Saint-Ouen Cedex Tél.: (1) 49 48 50 85

Service lecteurs n° 2 (dernière page)